AD_____

Award Number: DAMD17-98-1-8039

TITLE: FACTS (Find Appropriate Clinical Trials) for you: A Computer-Based Decision Support System For Breast Cancer Patients

PRINCIPAL INVESTIGATOR: Lucila Ohno-Machado, M.D., Ph.D.

CONTRACTING ORGANIZATION: Brigham and Women's Hospital
Boston, Massachusetts 02115

REPORT DATE: May 2002

TYPE OF REPORT: Final

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 2002	Final (20 Apr 98 - 19 Apr 02)

4. TITLE AND SUBTITLE

FACTS (Find Appropriate Clinical Trials) for you: A Computer-Based Decision Support System For Breast Cancer Patients

5. FUNDING NUMBERS

DAMD17-98-1-8039

6. AUTHOR(S)

Lucila Ohno-Machado, M.D., Ph.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Brigham and Women's Hospital
Boston, Massachusetts 02115

E-Mail: machado@dsg.harvard.edu

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. Abstract ***(Maximum 200 Words)*** *(abstract should contain no proprietary or confidential information)*

We have built a prototype of a web-based system that matches patients to NCI-listed clinical trial eligibility criteria. The system consists of a protocol encoder, an expression evaluator and inference engine that can handle uncertainty (via Bayesian belief networks), and user interfaces customized for the patient and the health care provider. We use an object-oriented model and standard nomenclature (UMLS) in this system. We use measures of value of information in the dynamic creation of data entry forms.
We have completed a formative evaluation of the system, by comparing its ranked list of trials to those of two oncologists for the same retrospective cases of stage IV breast cancer collected from the Brigham and Women's Hospital. We have also elicited the oncologists' impressions of the quality of recommendations given by the system. Although the number of cases and protocols was limited in this experiment, the results suggest that systems such as this one can select appropriate trials for a given patient, even when presented with incomplete or uncertain information.

14. Subject Terms (keywords previously assigned to proposal abstract or terms which apply to this award)	15. NUMBER OF PAGES 89
Decision Support System, Bayesian Belief Networks	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

Table of Contents

Introduction

This is the final report for this grant. The main accomplishments of every year are outlined in the body of this document. Year 4 was approved for completion of the work with no additional funds.

The purpose of this project was to build and evaluate a computer-based decision support system to help patients and primary care providers seek appropriate trials for their specific situation, even in conditions of uncertainty (missing data). The rationale for building this system was as follows: Although participation in clinical trials has been shown to improve health outcomes, accrual of patients is difficult and is estimated to be below 5% of the eligible population. Lack of information and automated tools to search clinical trials appropriate for each particular patient are some of the main reasons for low accrual. We built an inference engine for clinical trial eligibility that searches trials listed in the PDQ database of the NCI and ranks the trials that best fit a given patient, under conditions of uncertainty.

Body

Section 1. Overview of Tasks

We have proposed to build our computer-based eligibility determination engine in two stages: (1) build an ad-hoc deterministic (i.e., non-probabilistic engine not able to deal with uncertainty or consider associations among eligibility criteria and patient data values), and (2) build a probabilistic engine, based on belief networks, that is able to statistically infer values for missing data, given the information it can gather from the patient or health care provider, and can take into account associations among variables and patient data values.

A description of the research accomplishments associated with each Task outlined in the Approved Statement of Work (in **bold** face) follows:

***Task 1.* Analyze, structure, and construct data entry forms for eligibility criteria derived from clinical trials for breast cancer treatment available in PDQ**

a. **PDQ clinical trial summaries for health care professionals will be dissected**

We have created an explicit data model for the representation of criteria. This model is scalable and is based on standardized vocabularies. Please refer to Appendix 1 for more details.

b. **A structured format for storing eligibility criteria in a relational database will be defined**

We used the XML to structure and store eligibility criteria. A relational database was not necessary to store the eligibility criteria, as the XML files were deemed more general and could be parsed in real time with no performance degradation.

c. **WWW-based data entry forms will be constructed an linked to database**

Separate forms to address the needs of patients and primary care physicians were designed.

d. **Database for interim storage of patient data will be constructed**

XML files were used for this purpose.

***Task 2.* Construct simple models that do not model uncertainty to assess the need for belief network models:**

a. **Simple rule-based system construction using knowledge from domain expert**

We built two modalities of rule-bases systems. The first one did not incorporate the notion of uncertainty. In the second one, the outcomes of the rule-based system were updated to include probabilities of a criterion being met by a particular patient.

b. **Preliminary evaluation of simple rule-based system**

A comparison of system's performance with and without the probabilistic feature was made. We did not identify major differences in the outcomes of these models. Since this was a limited experiment that incorporated probabilities in an ad-hoc fashion, however, we were not sure whether more general conclusions about the usefulness of adding the probabilistic feature are warranted. We therefore proceeded a model based on Bayesian networks for this purpose.

Task 3. **If results from Task 2 show that belief networks are needed, construct belief network to model uncertainty in most common eligibility criteria and perform inference on entered data, else refinement of simple models and interface construction will take place:**

a. Belief network model will be constructed using knowledge from domain expert

We built two types of belief networks. One of them included very complex networks with several arcs. This type of networks, constructed by Dr. Huan Le, an internist post-doctoral fellow, was deemed inappropriate given the need for hundreds of values for the conditional probabilities, and its non-scalability and difficult maintenance. Dr. Nachman Ash, an internist postdoctoral fellow, constructed simple belief networks featuring relations among laboratory values that were frequently encountered in eligibility criteria. The belief networks dealt with demographic data and laboratory values related to liver, renal, and hematologic function.

b. Belief network model will be integrated with WWW and database environments to create application

We used two different belief network engines for the development of this system. The belief network engine used in an initial version of the system was built with Netica. The final one was based on JavaBayes and was shown to be more flexible and robust.

c. Algorithm for ranking possible trials for a patient will be implemented

Two ranking algorithms were developed for this project. Dr. Samuel Wang was responsible for the initial implementation, which had a great dependency on the number of values related to uncertain criteria. A new ranking algorithm was developed and implemented by Dr. Ash, in which other factors were considered. Details of the latter are given in Session 3.

d. GUI for displaying results and linking to specific summaries in PDQ will be built

Two graphical interfaces have been developed as the system evolved. The second one separated patient and provider interfaces to facilitate navigation.

Task 4. **Redesign of evaluation methods and interim analysis and system refinement:**

a. Evaluation methodology will be redesigned

The evaluation strategy was redesigned to conform to the realities of the clinical services at Brigham and Women's Hospital (BWH) and Dana Farber Cancer Institute (DFCI). The major consultants during the construction of the system were Dr. Ursula Matulonis and Dr. Darrel Smith. Dr. Craig Bunel also played a consultant role. The need for unbiased oncologists to properly implement the proposed clinical trial was the critical point for its implementation in year 3. These oncologists were identified and participated in the evaluation of the system. Retrospective data from Brigham and Women's Hospital was obtained for preliminary testing of the model, with filing and approval from the Institutional Review Board.

b. Interim analysis of the system using abstracted cases will be conducted

These cases were constructed based on actual retrospective data collected from the Brigham and Women's Hospital. Data from 20 patients admitted to Brigham and Women's Hospital with a diagnosis of breast cancer stage IV was used for thorough evaluation of the system and comparison of performance to that of oncologists. The items collected corresponded to those on the WWW forms and were collected from the electronic medical record. Dr. Ronilda Lacson collected these data, with assistance from Ms. Debra Delatorre.

c. System will be refined in terms of belief network model and GUI given interim analysis results and internal user feedback.

The initial implementation was completely substituted given problems with its performance and connectivity to the other components of the system.

Task 5. **Subject recruitment, abstraction of medical records, and creation of survey instruments for final analysis:**

a. Lay people ("patients") will be recruited

We have contacted used personnel from our group to serve this role. Although the groups contains physicians, it also contains administrative assistants, programmers, computer scientists, and interface design professionals who do not have medical training.

b. Medical records will be abstracted and randomized

Medical records were collected and abstracted by an internist. The data originated from the electronic medical record at BWH.

c. On-line forms for recording selection of clinical trials for patients and providers will be built

The construction of these forms was deemed unnecessary as the system underwent significant and frequent updates given the feedback from the users.

d. Surveys for assessing patient and provider satisfaction with the system will be built

The overall satisfaction with the system was good, although the survey was not formal.

e. Primary care providers and oncologists will be scheduled for final experiments

We have used two oncologists and one internist for secondary and primary evaluation, respectively.

***Task 6.* Evaluation experiments:**

a. Oncologists will assess system's performance

Details of the assessment can be found in Session 3.

b. "Patients" will use the system and fill on-line forms and surveys

On-line forms were filled by the users. The compliance with on-line surveys was minimal, hence individual informal surveys were conducted.

c. Primary care providers will use the system and fill on-line forms and surveys

On-line forms were filled by the providers. We did not attempt to conduct on-line surveys given the minimal compliance from the other users (item 6c).

Task 7. **Final analysis and report writing:**

a. Final analyses of data from oncologists, "patients," and providers will be performed

A detailed document of the system and its evaluation can be found in Session 3.

b. A final report and manuscripts will be prepared

This is the final report. An article was accepted and presented at a regular session and the student paper competition session at the American Medical Informatics Association Meeting in Washington DC, November 2001.

In the next sections, we describe the evolution of FACTs, and illustrate the description with some screen samples from the system.

Section 2. The initial FACTS

An overall summary of the goals and accomplishments for the first year of this project is given in [1]. The initial prototype is described below.

2.1. Design

FACTS utilizes an evaluation engine called EV to interpret Arden statements and expressions, including logical and temporal criteria. EV uses a lexer generated with flex 2.5.4 and a parser generated with Bison 1.25. Information about the clinical trial protocols, including the encoded criteria, is stored in XML documents. A separate XML parser is used to obtain the portion of an XML document containing the criteria encoded in Arden. Then the EV parser constructs an abstract syntax tree representing Arden statements and expressions that can be interpreted by invoking its "Evaluate" method. The evaluation of the abstract syntax tree follows an interpreter design pattern to recursively request the objects representing the nodes of the tree to interpret themselves and yield the result of the evaluation. The UML class diagram in Figure 1 illustrates the object-oriented structure of EV. Statements and expressions are related by inheritance to allow their participation in an interpreter or visitor design pattern.

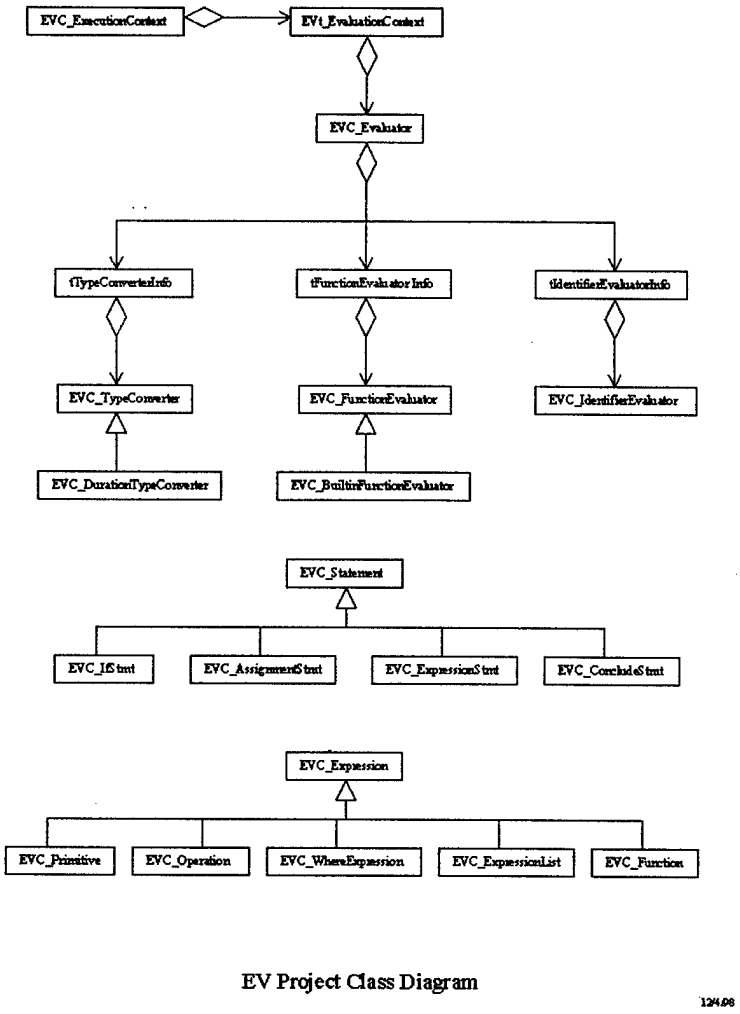


Figure 1. Project Class Diagram. Statements and expressions.

The UML class diagram in Figure 2 also illustrates the object-oriented structure of FACTS. The information in the encoded criteria is maintained by the criteria store object. Arden variables may be evaluated upon demand with the variable evaluator object. Identifier evaluators, function evaluators, and type converters may be registered with the EV evaluator object, which it will consider using in the course of evaluating a statement or expression.

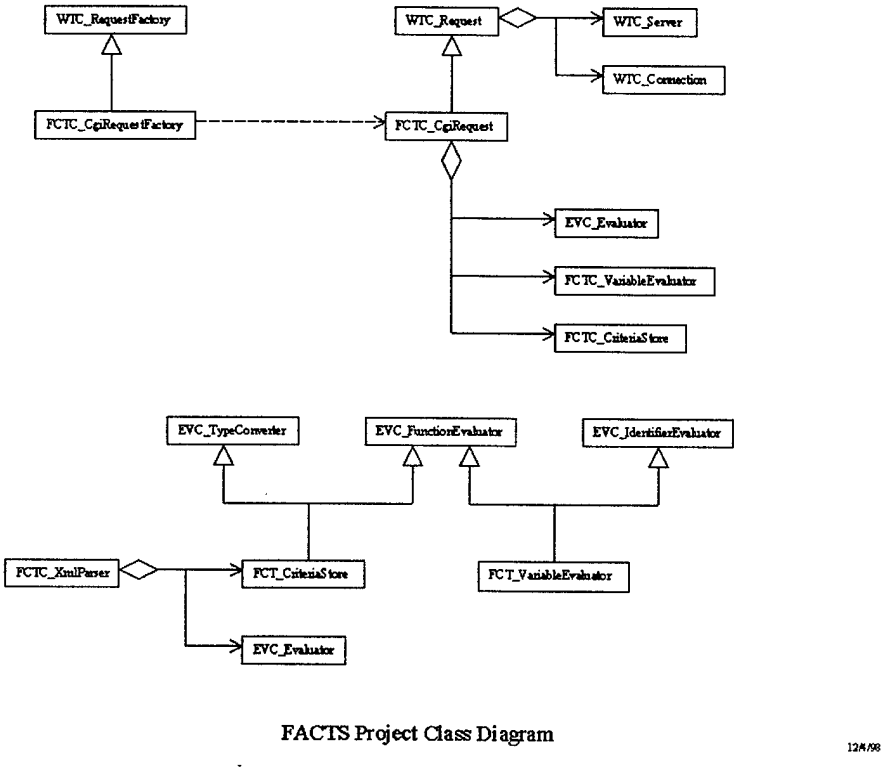


Figure 2. Project Class Diagram: Eligibility Criteria.

With regard to evaluating functions, EV provides a base class called EVC_FunctionEvaluator that may be derived from in other projects. These may be registered with EV to be potentially used in evaluation. In the EVC_Evaluator::EvaluateFunction method, the "Evaluate" methods of evaluators pointed to by elements in fFunctionEvaluatorSeq are invoked, starting with the last EVC_FunctionEvaluator that was registered and working backward, until an evaluator is found that does not yield an unknown error or the first evaluator in the sequence is reached. If a suitable evaluator is found, its return value is returned. If no suitable evaluator is found, this method yields a lookup error.

With regard to obtaining values of identifiers, EV provides a base class called EVC_IdentifierEvaluator that may be derived from in other projects. These may be registered with EV to be potentially used in evaluation. In the EVC_Evaluator::GetIdentifierValue method, if an identifier is not known in the immediate context, the identifier evaluators in fIdentifierEvaluatorSeq

are searched in reverse order, beginning with the last one to be registered. A particular evaluator is asked to determine the value of the identifier by calling the Evaluate method. If no error is generated, the identifier is considered to have been found. If there was an unknown error or lookup error, then searching continues. If there was another type of error, the routine fails. If after these lookup attempts the identifier has still not been found, the routine signals an lookup error.

With regard to evaluating sentences in DSG Arden in the form of an abstract syntax tree, EV mostly uses the interpreter design pattern. Work has been done on extended the capabilities of EV to use alternative evaluators for "where" expressions. The visitor design pattern is being implemented to accomplish this.

2.2. Development Retrospective

The FACTS project was initially developed to run on a UNIX server. Subsequently, it was modified to run on a Windows NT server. In 11/98, the variable counting algorithm used in FACTS was modified slightly. The variable counting algorithm in FCTC_CgiRequest::TallyVars was formerly the following.

The score for a particular variable is the number of criteria that are not definitely known that the variable appears in over the protocols that have not been probably ruled out or definitely ruled out.

This algorithm was changed to the following.

The score for a particular variable is the number of protocols that: 1) have not been probably ruled out or definitely ruled out and 2) the variable appears in within a criterion that is not definitely known.

The former algorithm allowed a particular variable to be counted multiple times in one protocol, whereas the new algorithm limits the count for a particular variable to one per protocol. The two different algorithms can produce different results, especially when a protocol specifies a variable in

multiple criteria. The most salient example of this that was found during testing involves the variable "metastases_locations". There are some protocols in which this variable occurs several times.

In 11/98, a variable constraining strength algorithm was implemented. As a preliminary measure, the variable ranking algorithm was refined to include a constant weight for each variable. The weight should satisfy

$0 \leq \text{weight} \leq 1$

and may be included in the XML file where any particular variable is described. The default variable weight is unity. The basic notion behind the weight is that it is the overlap between subpopulation prevalence and protocol disqualification.

The measure of the degree to which an unknown variable has the potential to rule out additional protocols may be called the "rule-out power" of the variable or the "constraining strength" of the variable (how strongly the variable constrains the set of operative protocols). The constraining strength s_i of the ith variable is

$s_i = F_i * w_i$

where F_i is the frequency of the ith variable (the fraction of the protocols that have not been probably ruled out or totally ruled out that the variable appears in), and w_i is the weight of the ith variable.

Changes to FCTC_XmlParser::ParseVariables were made to incorporate the ability to parse variable weights. Changes to FCTC_CgiRequest::TallyVars were made to perform the computation of the constraining strength for each tallied variable. Changes to FCTC_CgiRequest::PrintResults were made to display the variables of interest in ranked order. Class definitions were augmented with

additional data members and header files with additional type definitions as needed to track the additional information.

In 11/98, the server code was converted to an ActiveX object, and an Active Server Page was used to invoke the ActiveX object and dynamically generate the HTML document presented to the client as the result of a FACTS search.

In 12/98, some of the error handling was optimized for use under ActiveX. The reporting of warnings to the browser under the ActiveX object project has been enabled for the parts of the code that use FCTC_CgiRequest::fWarnings or FCTC_XmlParser::fWarnings, either directly or indirectly. Not all such handling of errors actually output warnings; some of the mechanisms used were incompatible with the recent change to ActiveX. The capability of the function ErrorText in the file FCT_Request.cpp has been expanded to explicitly handle several additional error codes. This should improve the specificity of reporting warnings.

In 1/99, some minor operator name changes were effected to increase compatibility with Arden. Formerly, the "and" operator had an alternative name "&&", and the "or" operator had an alternative name "||". This is no longer the case. Now the "and" operator has an alternative name "&", and the "or" operator has an alternative name "|". This was done to avoid conflicts with the Arden concatenation operator "||".

In 3/99, changes to the Arden interpreter were made to enable enhancements to the "where" operator in DSG Arden. Inheritance relationships among enum values was already supported previously in the FACTS code, and the "is-a" operator works on them. An allowance for parents of a FACTS data type (as opposed to value) was made at this time to enable inheritance relationships among struct fields.

The behavior of the "where" operator in FACTS has been modified so that the left argument of the "where" operator is expanded to include all hyponyms (descendants) of all items in that argument. If an item in the left argument is a member of a struct which has inheritance relationships to other

structs, then the list formed from the left argument is expanded to include also the corresponding members of all hyponyms (descendants) of the struct.

There was a requirement that the code to accomplish this alternative interpretation of the "where" operator reside in the FCTL project and not the EV project. This has been done so that EV does not know about this code specifically but will call this code when appropriate. This involved changes primarily to FCTL and also to EV, but the changes to EV were basically for defining the base visitor class only. The default behavior of EV is unchanged. In other words, these changes to EV are backward compatible with previous versions. The actual use of EV is the same.

The mechanism by which the enhancements operate is essentially that the visitor design pattern is used instead of the interpreter design pattern, which was used in the relevant parts of EV previously. A pure visitor design pattern was not used because it would result in changing the interface to abstract syntax trees in EV, and hence existing code that uses EV could not be easily reconfigured to take advantage of new features of this type. The interface could also be expanded later if desired.

EV contains the base class for visitors. Derived visitors may be defined in other projects to provide alternative interpretations of DSG Arden abstract syntax trees constructed in EV representing DSG Arden sentences (statements or expressions). Access to the operative visitor (of base type EVC_Visitor) is controlled by a configurable singleton (of type EVC_VisitorSource). Basically, to use a different visitor that provides an alternative interpretation, you would only need to reconfigure the "visitor source" with your visitor. Then the rest of the code that uses EV can be used in the same way, but your visitor will be used for the interpretation instead. A visitor is configured by invoking the SetVisitor method on the EVC_VisitorSource object.

The EVC_Visitor class provides the basis for a visitor design pattern to interpret the DSG Arden abstract syntax tree constructed by EV. In the visitor design pattern, a node (generally an object) in an abstract syntax tree is interpreted (evaluated/executed) by an outside object (the visitor). This is in contrast to the interpreter design pattern, in which the node itself contains the interpretation logic. With the interpreter design pattern it is difficult to extend the way interpretation is done, because

the application logic that accomplishes the interpretation is hard-coded into the nodes themselves. With the visitor design pattern, it is relatively easy to extend the way the nodes are interpreted; since the application logic that accomplishes the interpretation is put in a separate class, a new class can be derived that carries out the alternative interpretation.

The reason for using a visitor at this time is to allow a different interpretation for a "where" expression without putting the application logic for the alternative interpretation in the EV project. Specifically, the FACTS project has a need for interpreting "where" expressions in a way that makes use of information about inheritance relationships in the data model used in FACTS. In the future other projects can also implement their own interpretations by deriving a visitor subclass. The screenshot in Figure 3 shows the old version of the FACTS home page.

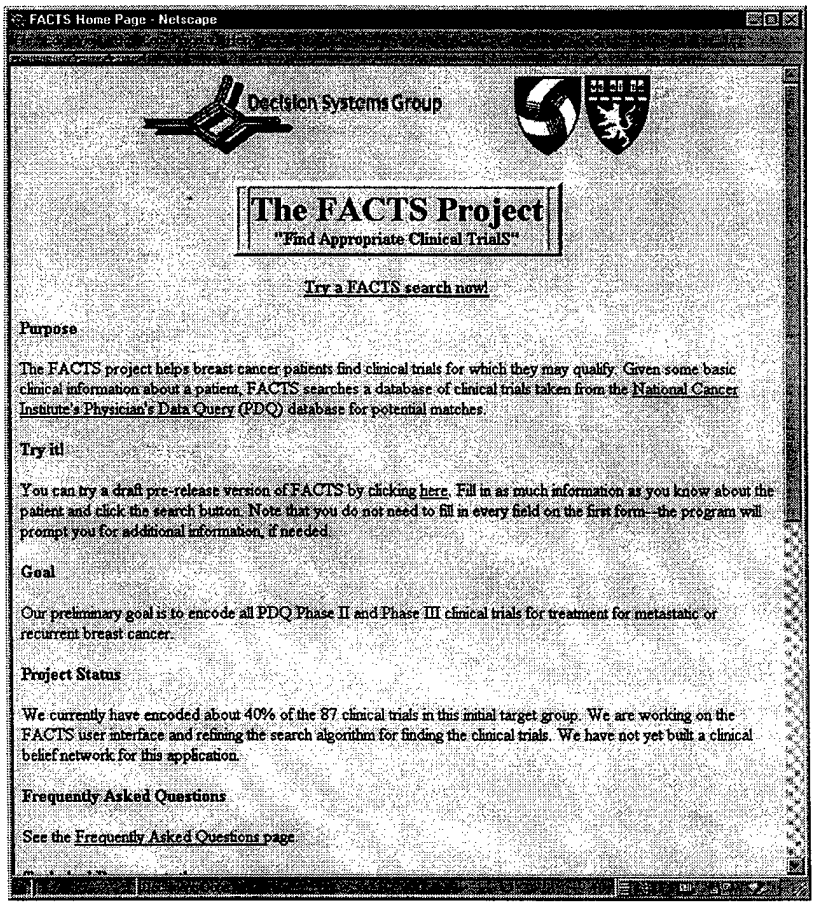


Figure 3. Initial Interface.

Figure 4 shows the old version of the FACTS search and results pages.

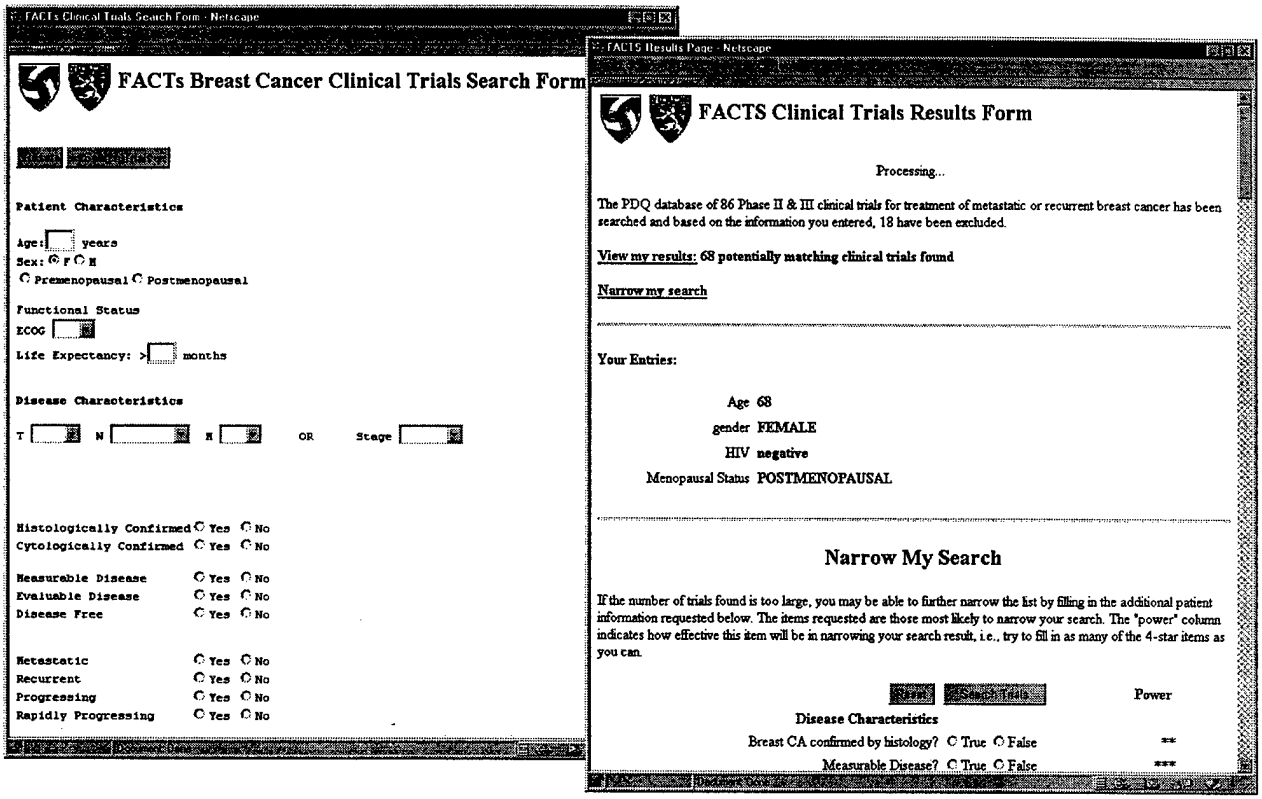


Figure 4. Initial search and results pages.

Figure 5 shows the subsequent versions of the FACTS Web pages.

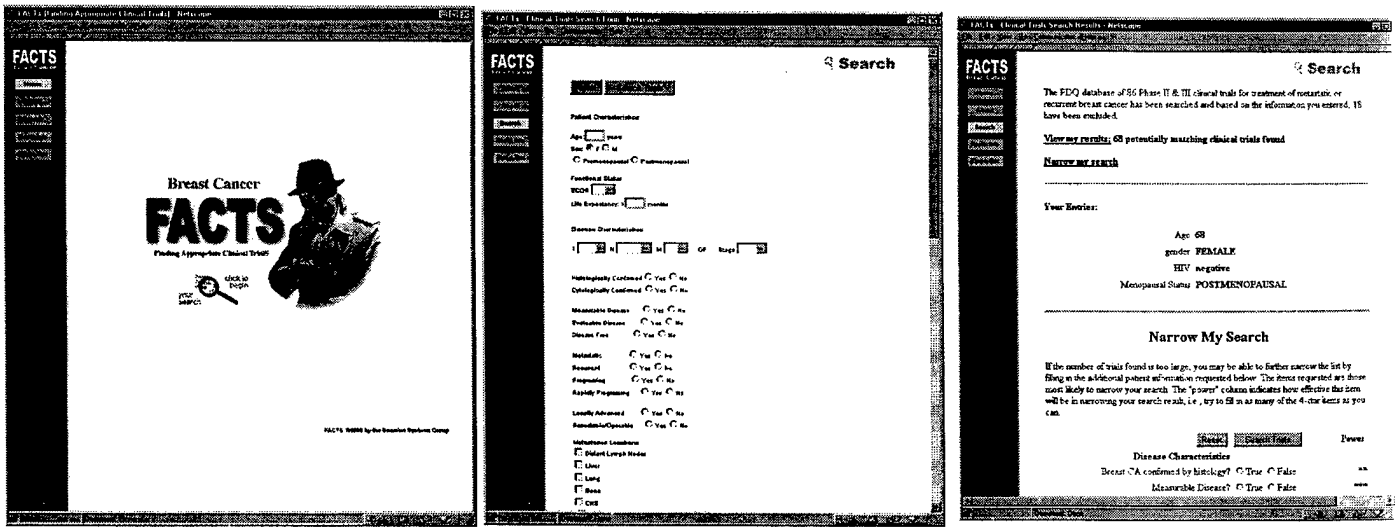


Figure 5. Some other versions of the interface.

Section 3. The New FACTS

The initial version of the system was substituted by the one described below.

3.1. System requirements

Final system requirements were outlined based on the goals of the FACTS project and previous experience with the initial prototype.

The system should:

- Collect patient data and return a list of clinical trials for which the patient may be eligible. Trials in which at least one of the entry criteria is not met should be filtered out.
- Rank the trials by the likelihood of patient's eligibility.
- Reason with any amount and content of patient data, inferring values for missing data.
- Adhere to and make use of standards in medical informatics (e.g., controlled terminologies).
- Be generalizable: use common clinical trial protocols, and be expandable to different medical domains (not only the one that serves for prototype development).
- Be able to represent most of the eligibility criteria (at least 90%).
- Create a sharable encoded clinical trial protocols database.
- Be available to both patients and health professionals.
- Be accessible from anywhere (e.g., patient's home, clinician's office, inpatient ward).
- Have an intelligent user interface:
  - Ask for data and present results differently by the type of user: health professional or patient.

- Ask for data items in an iterative way: ask first for the most common data items in the encoded protocols, generate results, and then let the user decide whether to enter more data, and thus narrow the list of appropriate protocols, or browse the results as they are. If the patient elects to enter more data, ask her for the most important data items.
- Avoid redundancy (e.g., the system should not repeat questions about previously answered data items, it should not ask for stage of disease if it is known that the patient has metastasis).
- Generate explanations: show why a criterion was evaluated to true or false, and why a protocol was ranked the way it did.

3.2. Clinical trial protocols

Clinical trial protocols were taken from the NCI's PDQ database [2].

This source of protocols was selected since it is the most comprehensive resource on cancer clinical trials, which includes information about clinical trials sponsored by the NCI and others. Since one of the goals of this project is to create a general system, it makes sense to use a comprehensive source of protocols, rather than local institution-specific protocol database.

Another advantage of using PDQ's protocols is their availability on the Web through CancerNet in a single format that facilitates automatic retrieval of eligibility criteria by parsing the HTML protocol document.

As a start, analysis and testing were restricted to a subset of protocols: Phase II and Phase III trials for the treatment of metastatic or recurrent women's breast cancer. Working with this subset is initially warranted since it simplifies development, but the goal of creating a scalable system that could be applied to other domains needed to be considered as design decisions were made.

The selected domain is specific, but extensive:

- Breast cancer is the oncology domain that contains the largest number of clinical trials (201 listed in the NCI database as of April 2001).
- Patients with advanced disease would be more interested in seeking participation in clinical trials after exhausting traditional treatment venues.

- Phase II and Phase III trials are further developed than trials in other phases, and typically involve more patients.

Seventy-nine phase II and phase III protocol trials for the treatment of metastatic or recurrent women's breast cancer were found in the NCI's database as of February 2001 (82 on April 2001).

3.3. Implementation

The system was redesigned in Year 3 to follow several principles:

- Medical knowledge was encapsulated in an object-oriented data model.
- Concepts were represented using standard vocabularies.
- Eligibility criteria were encoded in a logical expression language derived from Arden syntax.
- Bayesian networks were incorporated into the system's evaluation process for inferring missing patient data.
- Evaluated protocols were ranked by the likelihood that the patient might be eligible for each of them.
- The system had a platform-independent implementation based on Java.

The following sections describe the implementation in detail.

3.3.1. High level design

The system is designed as a thin client, server-based application (thus, computing power and storage are centralized on the server, not the client). The user accesses the application via the Web. The design is based on a viewer-controller-model paradigm. The viewer is composed of several Java Server Pages (JSP), which constitute the user interface. The controller is responsible for coordinating the flow of data between the user interface and the model, and is implemented as a Java servlet. The model is the heart of the application where the eligibility criteria are evaluated.

Figure 6 illustrates the architecture of the system. The data collected from the user interface are stored and processed in the data model object. The belief network infers additional values. The processed variables and their values are sent to the evaluator manager, which coordinates the

evaluation of the eligibility criteria. It takes criteria from the coded protocol database, and sends them with the appropriate data to be evaluated by the logical expression evaluator. The result of the evaluation of all protocols is the basis of a protocol's selection and ranking, which is presented to the user.

The "medi n the medical vocabularie

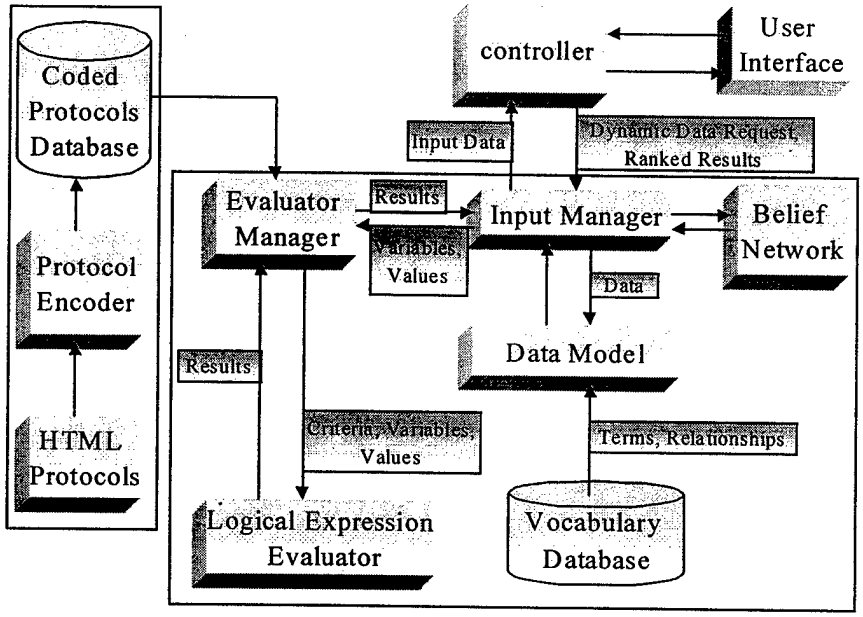


Figure 6. High level design of the new FACTS system.

3.3.2. Data

In order to achieve the goals of the project, mainly encoding most of the entry criteria, the data model of the system had to be extended. The approach used in the previous implementation of the FACTS project was, unfortunately, difficult to extend as the data model was built as a data dictionary defined in an XML document. Extending this model would require entering all the data-types and terms that need to be used by the system (which would hinder extensibility and flexibility). Moreover, this data model was domain specific. Applying the system to a different medical domain would require creating a new data model, or extensively modifying the old one. Therefore, a different approach was chosen by creating a domain-independent object-oriented data model.

The use of an object-oriented approach has the following advantages:

- Modeling a complex domain such as eligibility for clinical trials requires compound classes (or data-types). Although an object-oriented approach is not the only alternative (frames could be used as well) it is well suited for this purpose.
- The compound data-types of the old model could easily be transformed to objects with attributes.

- Inheritance plays a key role in creating a model that is easily expandable. For example, in the FACTS system data model BREAST CANCER is a subclass of CANCER. In order to extend the model to clinical trials in the domain of prostate cancer, all that is needed is to add a couple of new objects, PROSTATE CANCER PATIENT that extends PATIENT and PROSTATE CANCER that extends CANCER. These new objects will probably contain few attributes, since most of the needed attributes are inherited.
  - Inheritance makes it easy to construct the model (the same common attributes do not need to be rewritten).

The data were modeled based on analysis of the breast cancer protocols and the Common Data Elements (CDE) of breast cancer clinical trials developed by NCI [3]. The data items in the model are those required for determining patient eligibility for a clinical trial. The model was designed (using the Unified Modeling Language design tool by TogetherSoft [4]) based on common medical knowledge. Figure 7 illustrates the breast cancer model.

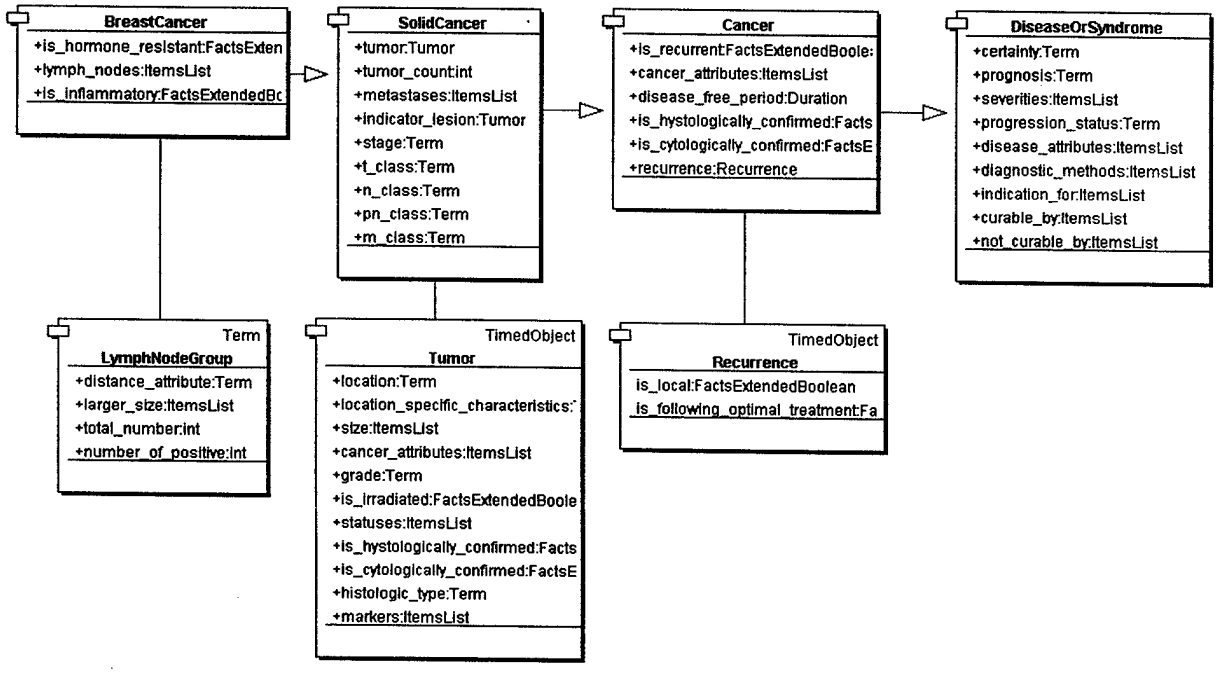


Figure 7. Part of the data model of breast cancer clinical trials.

The design of the model and the attribute names used in its classes impact the language created for encoding eligibility criteria (the variable names in this language are created by automatic transformation of attribute names – see discussion below). Therefore, it was important to use a design and names that resulted in "easily understandable" variable names. For example, the name of the histology type of the breast cancer tumor is represented by the variable name "breast_cancer.tumor.histologic_type.name".

Time plays an important role in evaluating eligibility for clinical trials. A frequent requirement, for example, is that certain treatment modalities had not been undertaken in a given time period ("more than 6 months since prior adjuvant chemotherapy"). Time was modeled by adding time stamps to data items (start_time, end_time and observation_time), and creating functions that use these time stamps to select the appropriate instance (latest, earliest, etc.).

It was also mandatory to model "not existing" in order to be able to say, for example: "the patient does not have congestive heart failure". That was done by adding an "is_present" attribute that is inherited by all objects in the model.

Patient data are stored in a model object ("BreastCancerPatient" in our case).

3.3.3 Use of standard medical terminologies

As opposed to the previous implementation of the system, the new system makes use of standard medical terminologies to represent terms and capture relationships between them. The advantages of using existing controlled terminologies are enormous:

- Time savings of not "reinventing the wheel": most of the needed terms and relationships already exist in standard vocabularies.
- A system that makes use of standard components is more acceptable.
- Terms in standard terminologies are mapped to the UMLS [5] and thus enable:
  - Linking of the system to other systems (like Electronic Medical Record systems).

- Using various terms and strings that represent the same concept (e.g. "CHF" and "Congestive heart failure" can be used interchangeably).
- Free text input is mapped to UMLS concepts, and thus gains a meaning.

Each term entered by the patient or used in the protocol eligibility criteria is looked up in the vocabulary database. The term's concept unique identifier (CUI) and its ancestors (terms which are more general in the thesaurus hierarchy than the patient's term) are retrieved, saved, and used while evaluating the encoded eligibility criteria (see Frame 1 for example).

Frame 1: An example of using CUI and relationships while evaluating

Text criterion: No history of diabetes mellitus
Encoded criterion: not have ("any name isa *diabetes mellitus* in diseases")

While the encoded criterion is evaluated the function "isa" checks if the value of the variable "diseases.name" isa "diabetes mellitus". That means that if the CUI of the value or one of its ancestors is equal to the CUI of "diabetes mellitus" the statement is evaluated to true.

Using relationships from standard terminologies has some pitfalls. The main one is that a terminology may contain hierarchic relationships that are inappropriate for the needs of the FACTS system. While generalization is suitable (e.g., "heart diseases" is a parent of "congestive heart failure), many other kind of hierarchic relationship are not. For example, in the COSTART vocabulary (one of the UMLS vocabularies), "diabetes mellitus" has a parent "Islets of Langerhans". While this relationship may be appropriate for the original intended use of this terminology, in the FACTS system the "isa" function may be inaccurately evaluated because of it. This problem was solved by restricting the use of relationships to two databases: MeSH (Medical Subject Headings) and Physician Data Query, giving priority to MeSH. These two were chosen because they contain most of the terms used in the clinical trial protocols, and appropriate terms' ancestors. For each term, the ancestors are taken from the MeSH database first. When there are no ancestors in MeSH, they are taken from the Physician Data Query database.

Some of the terms used by eligibility criteria in clinical trial protocols may not be found in the UMLS, and in some cases the necessary relationships may be missing from both MeSH and Physician Data Query databases. In that case, the user who encodes the criterion is able to add terms and relationships to the database.

3.3.4 Encoding language

Eligibility criteria are encoded using a variation of the Guideline Expression Language (GEL) [6], which is based on Arden syntax's logic grammar. Arden syntax was developed in order to facilitate sharing of medical logic among different health care institutions [7]. As the FACTS project is about using medical logic to evaluate eligibility for clinical trials, and since it is aimed at being sharable among institutions, the selection of the Arden syntax's logic grammar as the core of the encoding language was a natural choice. Moreover, Arden syntax was accepted as a standard of the American Society for Testing and Materials (ASTM) in 1992.

GEL was developed by the InterMed collaboratory (collaboration among medical informatics groups at Harvard, Stanford, and Columbia Universities [8]) for the GuideLine Interchange Format (GLIF) project [9,10] as a preliminary language that will capture the knowledge and logic of clinical practice guidelines. GEL differs from Arden syntax by letting the user define his or her own functions. This is a powerful property that enables extension of the language as shown below.

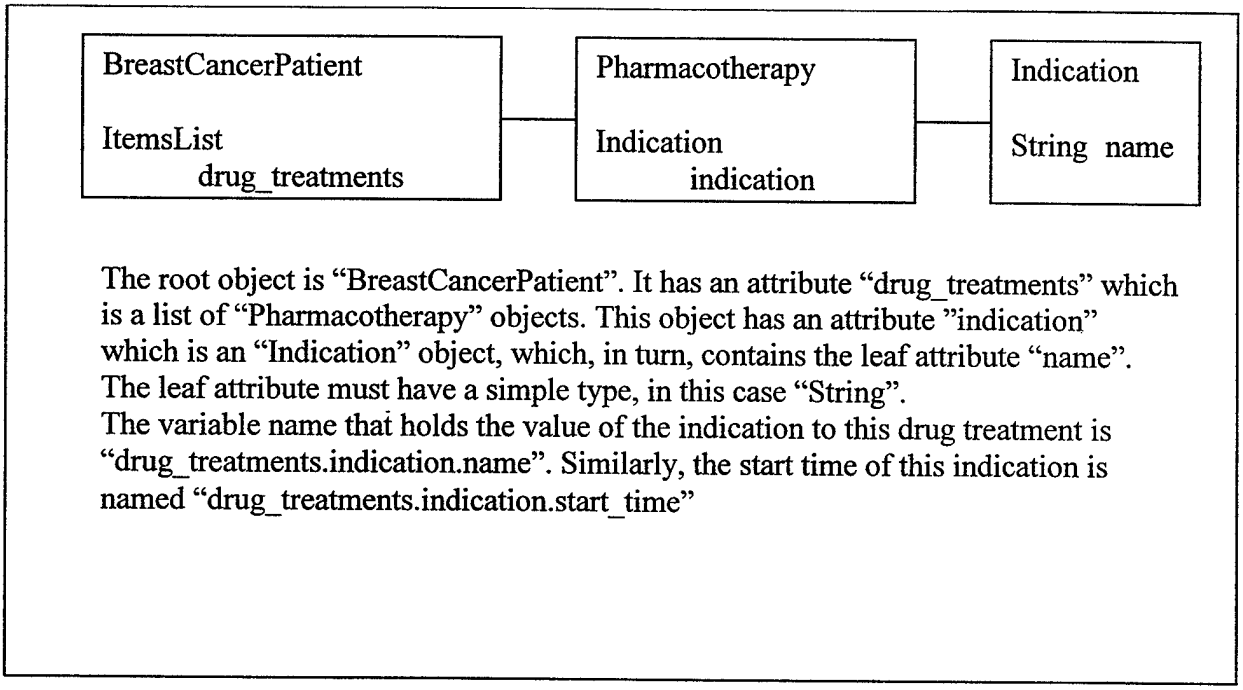
The encoding language is composed of 3 main components:

- GEL syntax
- Variable names
- Functions added to the syntax

The GEL syntax is a simple, yet powerful, logical expression syntax. It supports temporal functions and lists. However, it can deal with simple data types only (it supports neither complex data types nor objects). Therefore, the objects' fields in the data model need to be transformed into simple data type variables. This is done automatically by creating variables, the names of which are composed of the path of attributes from the root object to the leaf attribute (see Frame 2). The conversion function uses a depth-first search to create a total of 776 variables in the system.

Three functions were added to GEL for this project. Two of them (GET, HAVE) are used to retrieve values of variables from lists. These lists (of diseases, drug treatments etc.) contain complex data type (all attributes of disease or pharmacotherapy, for example). Since GEL does not support lists with complex data types, a function that retrieves the appropriate variable and sends it for evaluation is needed. The GET function gets the value of the variable, while the HAVE function checks if the requested item exists and returns an extended boolean (*true*, *false* or *unknown*).

Frame 2: Transformation of attributes in objects to variables with simple



The third function is ISA, mentioned above. It takes a variable name and a string, checks the variable value, and returns an extended boolean (for example, it returns *unknown* if the value of the variable is a parent of the string, such as, when the patient is known to have "heart disease", but the criterion is "not congestive heart failure" – it is unknown whether the patient's disease is congestive heart failure). The behavior of the function is complex, since it must take into account "no existing" values (the patient says that she doesn't have congestive heart failure), and components in a list (the patient says that she doesn't have any disease).

One of the goals of this work was to create a language that might be comprehensible to medical professionals who may encode their own trial's eligibility criteria. Limited by the syntax of GEL, functions were designed to take one long string argument that might be more comprehensible for reading than composite strings would be. This long string is parsed by specific functions. It contains

keywords that are used in various ways. Some of them indicate which item in a list should be retrieved (any, first, earliest, all, etc.), and others put constraints on the requested items (WHERE clause, CONTAINS clause). ISA can serve as a key word as well. NOTISA is another keyword, which is evaluated to not ISA.

As can be seen in the few examples given in Frame 3, the encoding language can be divided into two parts. The first one is retrieval of values from variables (GET and HAVE functions). The second one is a logical expression statement that is evaluated to *true*, *false* or *unknown*, and is the result of the criterion's evaluation.

Frame 3: Examples of encoded criteria.

Text criterion: Age 18 and over
Encoded criterion: age >= 18

Text criterion: Absolute neutrophil count at least 1,500/mm3
Encoded criterion: abs_neutrophil_count := get ("latest numerical_value from test_results where name
isa *NEUTROPHIL COUNT* and unit.name isa *cells/uL*");
abs_neutrophil_count >= 1500

Text criterion: At least 4 weeks since prior chemotherapy
Encoded criterion: had_chemotherapy := have ("any in chemotherapies");
chemo_end_date := get("ended_latest end_date from chemotherapies");
if had_chemotherapy then conclude not (chemo_end_date is within past 4 weeks); else
conclude not had_chemotherapy;endif;

3.3.5 Encoding process

The protocols selected for encoding were chosen by order of appearance in the search results of the PDQ database.

Encoding of the eligibility criteria is usually a manual process: each text criterion is examined and "translated" using an encoding language as described above. A special editor, created specifically for this project, retrieves the HTML page from the CancerNet™ Web site, delimits the eligibility criteria of that protocol, and presents them to the user, who needs to type in the GEL-based

encoding (Figure 8). If a criterion is already encoded, its GEL-based encoding is retrieved from the database.

Most of the criteria encodings are simple, but some are more difficult, and the result does not completely reflect the original text. Reasons include:

- Use of vague terms in the text criterion ("Adequate cardiac function" -- what is adequate? "Newly diagnosed disease" -- what is newly? Not treated? Time-related?)
- Deficiency of the data model for capturing some of the concepts ("No evidence of disease improvement by radiography" -- the model currently does not capture the method used to collect evidence).
- Avoidance of long and cumbersome encoded criteria ("...unless tumor involvement in treated or incompletely treated patients" -- although this expression could be encoded, it would make the criterion very long and confusing. In certain cases, keeping the criteria simple was preferred).

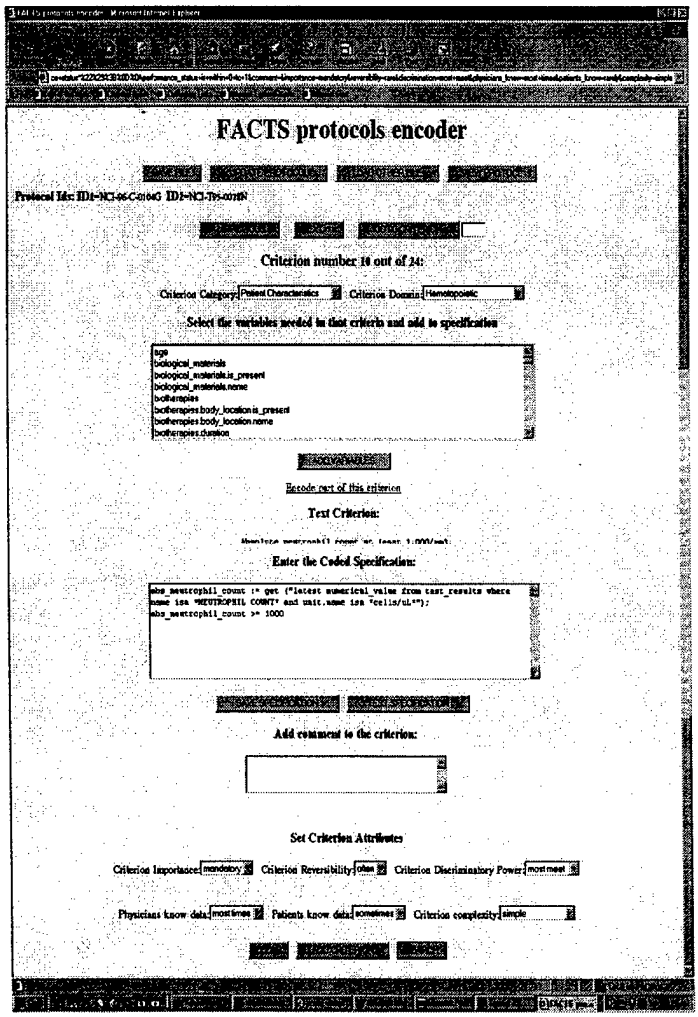


Figure 8. The FACTS protocols encoder. Text criterion is presented to the user who needs to type the GEL-based encoding in the middle window.

These difficulties were solved by different strategies:

- Transformation to a computable expression, even if not covering the whole semantics of the criterion (e.g., "Adequate cardiac function" is encoded by an expression that checks for normal ejection fraction).
- Use of vague terms in the encoded criterion ("uncontrollable hypertension") -- the user has to enter this information.
- Disregard of some information when it is considered not important (e.g., the method of measuring the ejection fraction is ignored with the assumption that most measurements are done by valid, interchangeable techniques).
- Addition of comments. The encoder can add comments that will be presented to the user of the system. The comment can clarify some aspects of the criterion, or just state that this encoding is not completely accurate.

The editor lets the user check the syntax of an expression for correctness, verify the legitimacy of variables' names used in the expression, and assess whether the terms used in the expression map to concepts in the UMLS.

For each criterion, the user needs to add the following information:

- The importance of the criterion (can it be ignored in some cases, or is it mandatory?).
- The reversibility of the criterion (if it is evaluated to false, can it change to true in the future?).
- Estimation of the discriminatory power of the criterion (do most patients who access the system meet this criterion? Or some of them? Or few of them?).
- Estimation of whether patients and physicians would know the values needed to evaluate this criterion (on a 1 to 5 rank scale).

This information is used by the system to rank the protocols and ask for more data (see below).

The encoded protocol is saved in both a Java object format (to be used by the system for eligibility determination) and an XML format (to view and share). Encoded criteria and information about the encoded protocols are saved in a relational database.

The time spent on encoding of each criterion is measured automatically and saved for analysis.

3.3.6. Missing data

The process of evaluating eligibility of a patient for clinical trials is data-intensive, as exemplified by the 776 variables defined in the system. Most users will probably enter only a small portion of the necessary values, both because they will not know the values of others, and because they will not be willing to spend sufficient time to enter all the required data. Therefore, it is expected that the system will have to deal with several missing values.

The new FACTS system infers missing values using two strategies. The first is deterministic: a missing value may be able to be deduced from a known value of a related parameter. The second is probabilistic and uses simple Bayesian networks.

3.3.6.1 Deterministic inference of missing values

There are two types of deterministic inference:

- Updates of linked data items using domain knowledge. For example: if a patient is known to have metastases, we know the stage of her disease (stage 4), or if a patient is known to be postmenopausal, she is also not pregnant, not fertile and not breast-feeding.
- Transformation of measurement units: different criteria may use different measurement units of the same test. For example, **ECOG 0-1** and **Karnofsky 70-100%** are two equivalent criteria regarding the performance status of a patient. When the system knows the value of the patient's performance status (in either measurement scale) it adds the value in all other possible scales. Thus any criterion using related measurement scales gets evaluated properly. This is used extensively for laboratory results that may be expressed in different units.

This kind of inference of missing values is important for several reasons:

- As the evaluation engine gets more information, its performance becomes more accurate, since more eligibility criteria are evaluated to a value other than *unknown*.
- It reduces the input burden: the system avoids asking the user to enter information on related items.
- Inconsistencies in input data are avoided.

3.3.6.2 Probabilistic inference of missing values

The protocol ranking may be more accurate by inferring missing values, since the ranking algorithm weighs results differently if they are based on inferred values (see below for more details). The system makes use of simple Bayesian networks to infer missing values.

A Bayesian (belief) network is a directed acyclic graph in which nodes represent variables, and arcs between nodes represent probabilistic relationships [11]. The network is created by selecting the desired variables needed to model the domain, adding appropriate causal arcs between them, and assigning prior and conditional probabilities. If some values of the variables are observed, the values of others can be inferred using Bayesian inference.

As discussed earlier, Bayesian networks have been proposed for eligibility evaluation systems by modeling the entire set of eligibility criteria of a protocol (or more than one) in a complex collection of networks [12,13,14]. This approach is not feasible for determining eligibility for multiple clinical trials. Therefore, creating several small independent networks that infer missing values of specific patient data items was preferred. These are general-purpose networks, modeling common medical knowledge related to frequently appearing data items in clinical trial protocols.

Currently, the system uses four separate directed acyclic graphs, representing age-related items (Figure 9), liver function tests, white blood cell counts, and pulmonary function tests. There are a total of 31 nodes in these graphs. The Bayesian networks were implemented using JavaBayes [15] as the Bayesian inference software.

Prior and conditional probabilities that populate these networks were taken in part from the medical literature (e.g., [16]). The remaining probabilities were estimated by the author based on medical knowledge. In the future, these probabilities could be updated by using relevant patient data, as they become available, in a manner suggested by Neapolitan [17]. Possible sources of such information may be clinical databases, and the database that will be created by data collected by the system.

The known patient data (data entered by the user) are inserted into the Bayesian networks as the observed evidence. The posterior probabilities are then calculated for all unknown variables in the network. If the posterior probability of a specific value is above a certain threshold (currently set to 5% above the chance probability), it is selected as the inferred value for the variable.

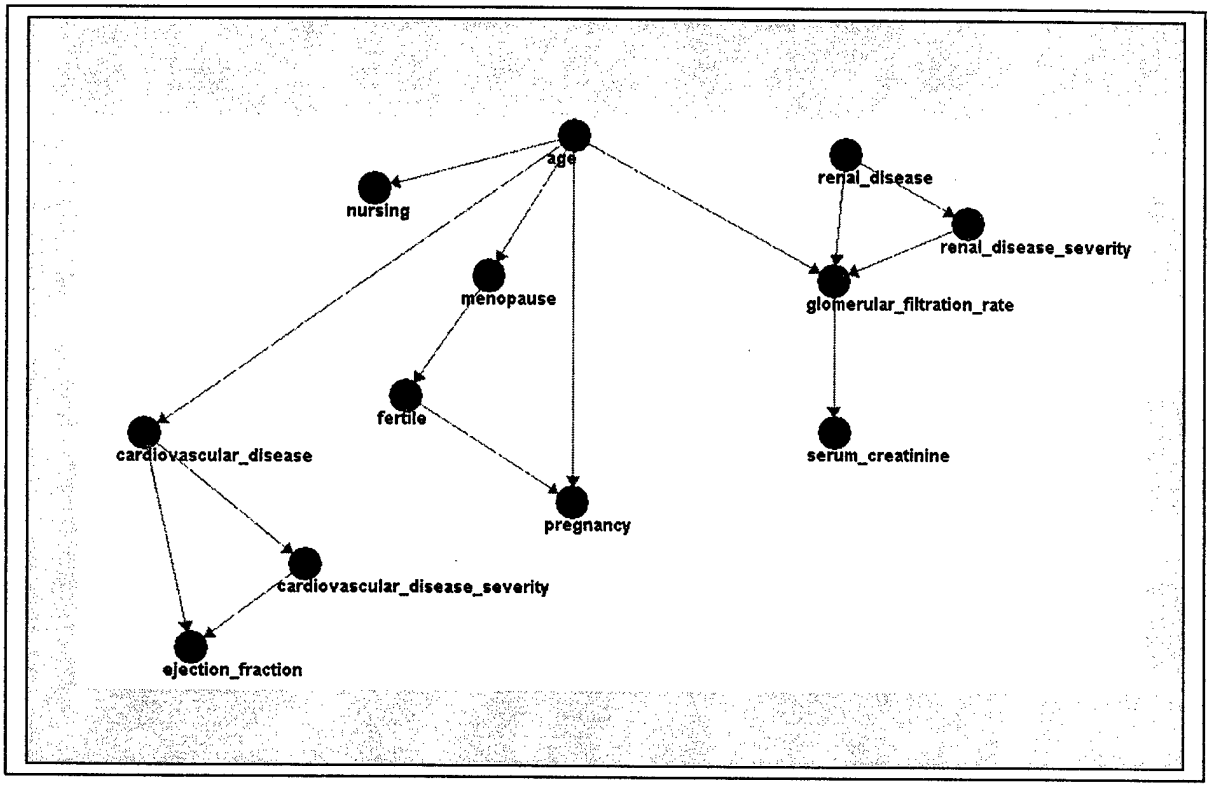


Figure 9. Age-related items organized in a typical Bayesian network used by the new FACTS

The posterior probabilities are not considered in the ranking of the protocols. Thus, a value inferred with a probability of 90%, and a value with a posterior probability of 30% (provided that it is above the threshold) are given the same weight during the ranking process. This limitation will be discussed later.

3.3.7. Evaluation of encoded criteria

A GEL parser / evaluator , built for use in the GLIF project (developed by Omolola Ogunyemi, Decision Systems Group, Boston, MA), evaluates encoded criteria. Variable names are replaced with values (if existing), and each expression in the criterion is evaluated. The evaluation result of the criterion is an extended boolean value (*true, false* or *unknown*). If the criterion can not be evaluated because of missing data, the result is *unknown.*

Each criterion is evaluated twice: once with data entered by the patient including deterministically-inferred data (definite data), and afterwards with probabilistically-inferred data. In the second round some of the criteria previously evaluated to unknown are evaluated to true or false.

The final result of a criterion evaluation is given as a letter symbol:

- **T** - criterion that evaluated to *true* based on entered and deterministically-inferred data only.
- **t** - criterion that evaluated to *unknown* based on entered and deterministically-inferred data, but evaluated to *true* when probabilistically-inferred data were added.
- **U** - criterion that evaluated to *unknown* based on entered, deterministically- and probabilistically-inferred data
- **f** - criterion that evaluated to *unknown* based on entered and deterministically-inferred data, but evaluated to *false* when probabilistically inferred data was added.
- **F** - criterion that evaluated to *false* based on entered and deterministically-inferred data only.

Thus, we get a rough qualitative measure of the likelihood that a patient meets the criterion: *T* and *F* represent the two extremes (100% and 0% respectively), and *t*, *U* and *f* represent ordinary intermediate values.

The result of a protocol evaluation is a list of these symbols, one for each criterion in the protocol.

3.3.8. Ranking of protocols

As stated above, the protocols should be ranked for a patient by the likelihood of that patient's eligibility. This is accomplished by examining and aggregating the evaluation results of the individual criteria in the protocol.

The patient is considered eligible for protocols for which all of the criteria evaluate to *T*. These are ranked highest and presented by the number of criteria that they contain.

Protocols for which one or more criteria evaluate to *F* are considered as inappropriate for the patient, and are therefore filtered out. Nevertheless, it is important to present these protocols to the user, and let him or her investigate why they were rejected. They are ranked separately, as discussed below.

The rest of the protocols contain any combination of criteria that were evaluated to *T*, *t*, *U*, or *f*. These are ranked by a weighted score that is dependent on the number of criteria that were evaluated to *t*, *U* and *f*. The weights represent the notion that the patient has a higher likelihood of eligibility for trials in which the criteria evaluated to *t*, than for those in which the criteria evaluated to *U* . Similarly, a higher likelihood of eligibility for trials in which criteria evaluate to *U* is expected than for those in which criteria evaluate to *f*. Criteria that evaluate to *U* are weighted by their discriminatory power, using a scale predetermined by the encoder (see in "encoding process", above). Thus, a criterion with higher discriminatory power (i.e., one that is believed a priori to be

true for only a small portion of breast cancer patients) gets a lower weight, and one that is believed to be true for most of the patients gets a higher weight.

It is important to notice that criteria that evaluate to *f* are not filtered out, but they have an increased probability of being ranked lower, determined by the weight of the criterion.

The algorithm described above was used to give each protocol a bottom line measure of appropriateness for a given patient on a scale of 1 to 5. Protocols for which all criteria evaluate to *T* get the maximal score of 5. Protocols for which at least one criterion evaluated to *F* get the minimal score, 1. Other protocols may get a score of 4 (the patient is probably eligible for the protocol), 3 (possibly eligible) or 2 (possibly ineligible), depending on the weighted score of the criteria, as described above.

As mentioned above, protocols that contain criteria that evaluate to *F* are filtered out, but are presented to the user for inspection. These protocols are ranked by the likelihood of the patient's eligibility despite this result (i.e., the protocol can be useful in the future if, for example, the patient's status changes, or if the clinical trial researcher believes that the criterion that evaluated to *F* is not too important). This ranking is achieved by evaluating the importance and reversibility scores that were given to the criteria during encoding (see above). If the criterion that evaluated to *F* is deemed not very important and is reversible, the patient may become eligible for the protocol. On the other hand, if the criterion is important or irreversible, then the patient is definitely ineligible for the protocol, and it will be ranked lowest.

Frame 4 contains a simple example of a ranked protocol list.

Frame 4: Example of ranked protocol list. The first one contains 1-*t*, 8-*U*, 1-*f*. The second one contains 2-*t*, 9-*U*, 1-*f*. Therefore, there is a higher likelihood that the patient is eligible to the first protocol that contains fewer unknown and probabilistically-inferred criteria. The two bottom protocols are filtered out, since they contain at least one criterion that evaluated to *F*. Notice that protocols containing criteria that evaluated to *f* are not filtered out.

```
protocol: NCI-G00-1878 ranked 1
20 criteria in this protocol were evaluated as follows:
  U  U  T  T  T  U  T  T  T  f  T  T  U  T  U  U  t  U  U  T

protocol: NCI-96-C-0104G ranked 2
24 criteria in this protocol were evaluated as follows:
  U  U  T  U  U  T  T  U  U  T  T  T  U  U  t  T  f  T  T  T
T  t  U  T

The following protocols are NOT appropriate for the patient:

protocol: NCI-G00-1834
24 criteria in this protocol were evaluated as follows:
  U  U  T  U  U  U  T  U  T  U  T  U  U  T  T  T  F  T  T  T
T  t  U  U

protocol: NCI-V97-1341
20 criteria in this protocol were evaluated as follows:
  T  U  U  T  T  U  T  T  U  T  f  T  F  U  f  t  T  U  T  T
```

3.3.9. User interface

The user interface was implemented as several JSP files that are controlled by a Java servlet. All pages, except the first introductory one, are generated dynamically, depending on which protocols are encoded, what input from the user is available, what the evaluation result of the protocols is, and what the user wants to see or do.

There are two user interfaces: one for use by patients and their representatives (herein called the "patient" interface), and another for use by health professionals. They differ in several aspects:

- The data items requested of the user (e.g., the patient is not asked to estimate her life expectancy, or to describe the histology type of her tumor).

- The way the request for data is presented to the user (e.g., when asked to enter the daily performance status, the patient gets a detailed description of the choices, while the health professional is asked to enter the value of the ECOG performance status).
- The way that the user enter the data (e.g., the patient is requested to enter diseases by using a simple menu, while the physician enters them as free text).
- The way the results are presented to the user (e.g., the patient gets a list of protocols for which she may be eligible, while the health professional gets also the evaluation results of the criteria, and the list of protocols that were filtered out).

The first input form refers to values of most frequent data items in the encoded protocols (Figure 10). The encoded criteria are analyzed automatically to find those that appear most frequently. For each data item, the program checks if there is no limitation on presentation to the user. Some items are not presented to patients either because they probably would not know the value, or for other reasons (e.g., life expectancy is too sensitive a topic for the patient interface).

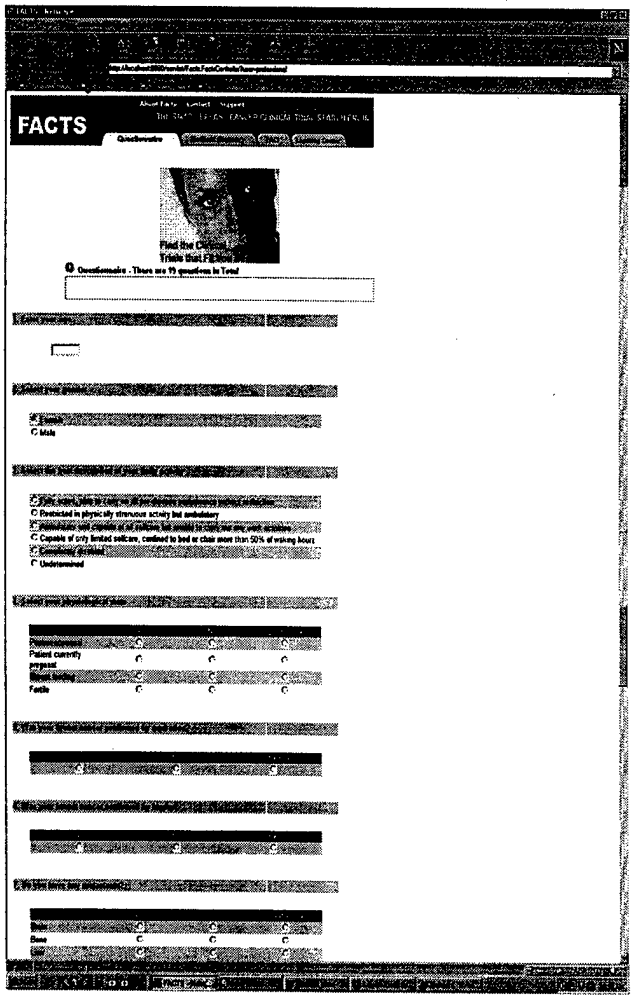


Figure 10. First input form generated dynamically based on the encoded criteria.

When the user submits her first set of answers, the system checks the data for allowed values, and evaluates the encoded criteria with the patient data. The user is presented with the number of appropriate protocols found, and can choose either to see the results or to enter more data in order to further narrow the protocol list.

Other input forms are created dynamically for data in criteria that evaluated to *unknown*. Once again, if the criterion is considered a priori as probably not known by the patient (as determined by the encoder of the criterion), it will not be asked. The system does not repeat questions for items that were already answered (even if they are still unknown).

The user may answer any item she wishes, and skip others. The system can reason with any number and content of data items.

The full results are presented to the user as a ranked list of protocol names. The clinical trial names are linked to the corresponding protocol summaries at CancerNet according to the type of the user (e.g., results for patients are linked to patient summaries).

Health professionals are exposed to a more detailed result (Figure 11), including the evaluation results for the criteria (the numbers of those that evaluated to each of the categories T,t,U,f,F), and protocols that were filtered out.

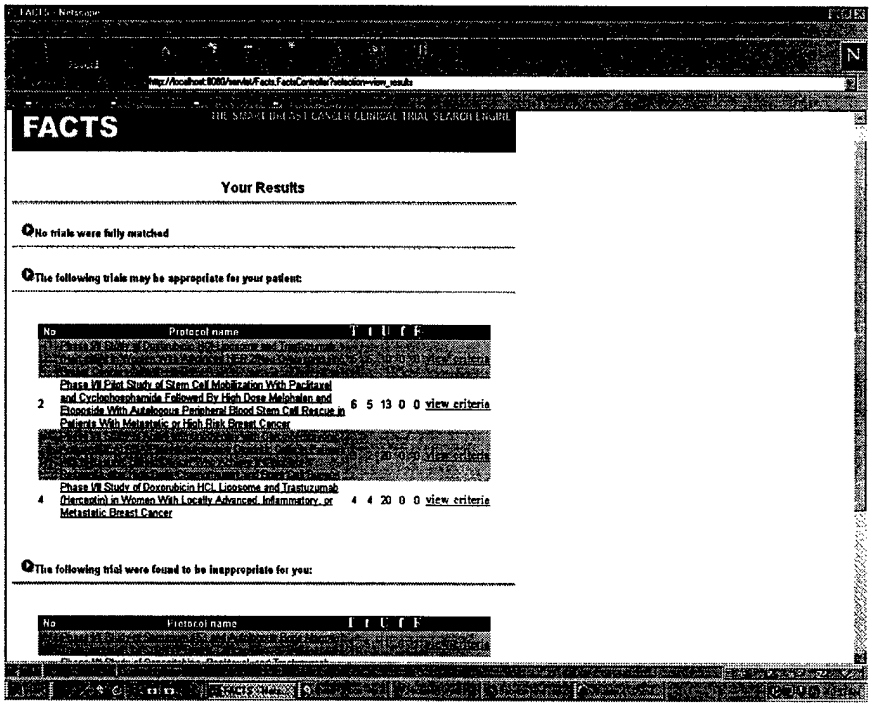


Figure 11. Presenting results to health professional: the names of the protocols presented with the number of criteria evaluated to *T*,

3.10. Evaluation

A preliminary evaluation of the system's selection and ranking algorithms was conducted, in order to get a preliminary measure of its agreement with selection and ranking by expert physicians. Patient data were abstracted from medical records of 20 patients with active metastatic or recurrent breast cancer, who were consecutively hospitalized during 1995 at the Brigham and Women's Hospital, Boston, Massachusetts. Forty-three data items were examined for each patient (items related to patient characteristics, disease characteristics, past treatment, other diseases and test results). Researchers not familiar with the encoding process and the particular encoded protocols collected the data. They decided which data items to collect by general familiarity with PDQ's protocols.

Two independent oncologists evaluated the appropriateness of the protocols for each of the patients, and ranked them. The physicians were given a short narrative description of the patients' data, and the

full abstracts of 10 protocols as downloaded from NCI's CancerNet Web site. When evaluating the appropriateness of the protocols for each patient, they were requested to give a score for each protocol (from 1 to 5, similar to the system's score, as described above), and then to rank the protocols that they found appropriate for the patient.

The system used the same patient data to evaluate the eligibility of the patients for each of the clinical trials.

The agreements on selection and ranking of protocols between the system and each physician and among the physicians were calculated using the kappa and weighted kappa statistics [18,19]. Statistical analysis was conducted using Microsoft Excel and Analyze-it [20].

Section 4. Results

4.1. Encoding process

The first 10 protocols listed on the search results from NCI's database were encoded. Each protocol contains between 20 and 41 eligibility criteria (mean 27.2). Out of 272 criteria, 228 (83.8%) criteria were unique. Criteria were considered unique if they were written in the protocols in a unique manner. If, for example, two criteria express the same idea, but are written differently, they represent two unique criteria (e.g., "No other concurrent antineoplastic agents" and "No other concurrent antineoplastic therapies").

It was feasible to encode 269 (98.9%) criteria. Thus, between 96.4% and 100% of the criteria in each protocol were encoded. The encoding process resulted in 141 (61.4% of the unique criteria) distinct encodings (in our example above, the two unique criteria had the same identical encoding).

Three criteria were not encoded. Two of them ("no prisoners" and a criterion related to a specific geographic location) lacked representation in the model. The third ("No other concurrent medical or psychological condition that would preclude study compliance") is difficult to encode because it involves complex human judgment. A total of 39 other criteria (27.6%) did not represent their text version with 100% accuracy (e.g., "No medical or psychiatric condition that would increase risk" was encoded as "No severe medical or psychiatric condition" -- since assessment of risk is subjective, it is difficult to encode for computation purposes).

A moderate number (30.3%) of the encoded criteria were lengthy (> 255 characters), which is indicative of their being among the more complex criteria.

Table 1 presents the encoding time for 77 criteria from the last three protocols. Approximately 20% of the criteria were labeled as difficult or complex. Retrieval of the code from the database was possible in 23.3% of the criteria, as these criteria were already encoded in other protocols. Most of the criteria were encoded in less that 4 minutes, but in some cases nearly one hour was necessary (this includes the time taken to make some changes in the data model in order to enable encoding of these criteria). The average encoding time was 5.88 minutes (median 2.1). Therefore, encoding an average-sized protocol may take about 3 hours.

Table 1: Average encoding time of 77 criteria stratified by difficulty.

Criterion Difficulty	Number of Criteria	Average Encoding Time (Min)
Automatic Coding	18	≈ 0
Trivial	8	1.47
Easy	35	3.52
Difficult	9	11.12
Complex	5	28.12
Very Complex	2	36.80

4.2. Preliminary system evaluation

Data from 20 patients with metastatic, locally invasive, and recurrent breast cancer were collected from medical records of the Brigham and Women's Hospital, Boston. About 25% of the 43 data items requested for each patient had missing values. Age distribution was 25-71 years (mean 44.4). Other patient characteristics are shown in Table 2.

Table 2: Patient characteristics.

Data Item	No. of patients (percent)	Data Item	No. of patients (percent)
Disease Stage:		**Known Metastases**	11 (55%)
Stage IV	5 (25%)	Liver	7 (35%)
Stage IIIb	5 (25%)	Lung	4 (20%)
Unknown	10(50%)	Bone	5 (25%)
Tumor Histology:		**Recurrent Disease**	3 (15%)
Invasive Ductal Ca.	1 (5%)		
Unknown	19 (95%)		
Confirmed Histology/Cytology	17 (85%)	**Locally Advanced Disease**	8 (40%)
Measurable/Evaluable Disease	14 (70%)	**Known Lymph Node Involvement**	9 (45%)
Menopausal Status		**Other Diseases:**	
Postmenopausal	5 (25%)	Hypertension	3 (15%)
Premenopausal	8 (40%)	NIDDM*	1 (5%)
Unknown	7 (35%)	Asthma	1 (5 %)
Past Treatment			
Chemotherapy	16 (80%)		
Radiotherapy	6 (30%)		
Biotherapy	8 (40%)		
Hormonal therapy	7 (35%)		
Surgery	7 (35%)		

*Non Insulin Dependent Diabetes Mellitus

Table 3: Distribution of criteria evaluation results.

Criteria Evaluation	Criteria Number (percent)
TRUE	2283 (41.96%)
FALSE	210 (3.86%)
UNKNOWN	2947 (54.18%)
true (inferred)	515 (9.47%)
false (inferred)	39 (0.72%)

The process of protocol selection for these 20 patients involved 5,440 evaluations of 272 criteria (each criterion was evaluated 20 times, each time with different patient data). As can be seen in table 3, about 54% of the evaluations resulted in *unknown* because of missing patient data. After inference by the Bayesian networks, 18.8% of these evaluated to either *true* or *false*.

The system selected from 1 to 9 protocols per patient (Figure 12). On average 3.05 protocols were selected per patient. None of the selected protocols received an appropriateness score of 5 (*definitely eligible*) or 4 (*probably eligible*), 25 were graded 3 (*possibly eligible*), and 36 were graded 2 (*possibly ineligible*).

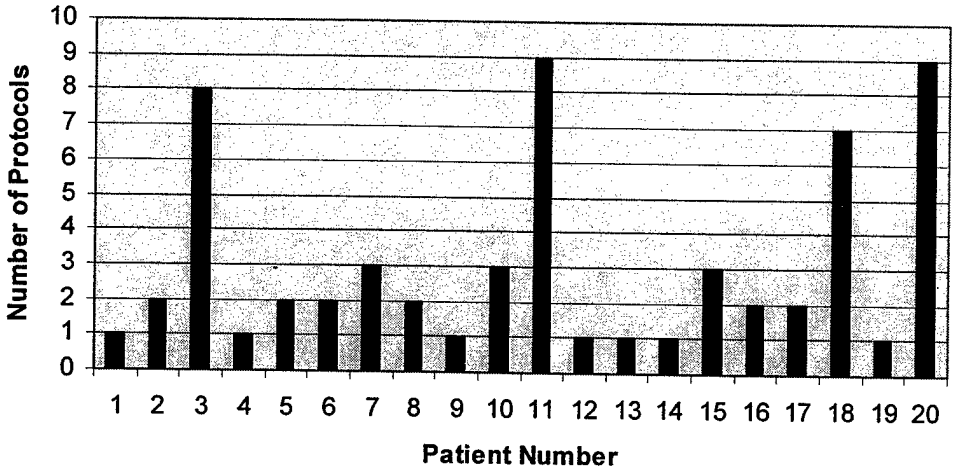


Figure 12. Number of protocols selected

In order to see the impact of inferring missing values by the Bayesian Network, the system was tested with and without Bayesian network inferred values. As expected, fewer protocols received grade 3 without the Bayesian network inference (19 without versus 25 with the probabilistic inference). The protocol ranking was affected for 4 patients. In two of them, the protocols ranked first and second were swapped as a result of adding inferred values.

The system's results were compared to physicians' selection of protocols with respect to two aspects: (1) the agreement on whether the patient would be eligible for each protocol, and (2) the agreement on protocol ranking for each patient. The kappa statistic for patient eligibility was 0.86 (95% CI 0.72 - 1.00) for one physician and 0.76 (95% CI 0.62 - 0.9) for the other. The agreement between the two physicians was 0.72 (95% CI 0.58 - 0.86).

The agreement on ranking the protocols was low: weighted kappa of 0.24 and 0.14 between the system and the two physicians respectively, and 0.31 between the two physicians.

4.3. Analyzing disagreement

There are two possible kinds of disagreement on selection of protocols: (1) the physician might select a protocol that the system found to be inappropriate for the patient (**extending disagreement**), and (2) the physician might not select a protocol that the system found to be appropriate (**narrowing disagreement**). There were 2 narrowing disagreements and 10 extending disagreements with one physician, and 14 and 6, respectively, with the other. Thus there were 16 disagreements of each kind altogether. The physicians shared only 4 of the disagreements (2 of extending type and 2 of narrowing type).

Table 4: Classification of disagreements between the system and the physicians.

Type of disagreement	Number of disagreements
Lack of model representation	1
Encoding mistake	1
Simple inference of missing value by physician	1
Complex inference by physician	12
Physician mistake	6
Interpretation of a borderline pathologic test result	3
Use of information other than eligibility criteria	1
Misinterpretation of patient data	3

In each case, the physicians were asked to explain their decisions. Based on the explanations, several common reasons for disagreement were found (table 4):

- **Insufficient model representation** causing inaccurate criterion encoding. For example consider the following inclusion criterion: "Previously treated with paclitaxel and an anthracycline (if medically appropriate) as adjuvant therapy or for metastatic disease". The encoding of this criterion checks if the patient got treatment with these drugs, but does not check if this treatment is "medically appropriate" for the patient (this was added

as a comment for the user). In one case, it was known that the patient did not get these therapies (and therefore the system evaluated the criterion to *false*), but one of the physicians considered these therapies inappropriate for the patient, and therefore decided that the patient met the criterion (extending disagreement).

- **Encoding mistake -** wrong code for a criterion.

- **Simple deterministic inference of missing value** – a physician deduced a missing value from another known value, while the system failed to do the same.
  For example, both physicians concluded that a patient with chest wall involvement is eligible for a trial that required locally invasive disease, while the system failed to infer that chest wall involvement implied locally invasive disease.

- **Complex inference of missing value** – a physician made some assumptions and inferred new information about the patient.
  For example, the physician inferred that a patient with metastatic, non recurrent and non progressive disease who received chemotherapy in the past, received it for treatment of the metastatic disease (and therefore was not eligible for a protocol that excluded patients with previous chemotherapy for metastatic disease).

- **Physician mistake**, usually as a result of ignoring some known information about the patient, or failure to notice a criterion in the protocol.

- **Interpretation of a borderline pathologic test result** as not clinically justifying exclusion from the trial.
  The system has a deterministic approach to test results: any value outside a limit specified by the criterion will result in evaluating the criterion to *false*. Sometimes physicians may disregard a result that is only slightly beyond appropriate limits. For example, one of the physicians decided that ejection fraction of 47% is appropriate even if the criterion required a normal ejection fraction (above 50%).

- **Use of information other than eligibility criteria** -- Physicians considered information given in the clinical trial protocol outside of the eligibility criteria section.
  For example, in one protocol, the title of the trial restricted the trial to patients with metastatic disease, but no corresponding eligibility criterion was stated.

- **Misinterpretation of patient data** resulting from unclear presentation of the case. For example, a patient with recurrent disease and skin involvement was considered by one of the physicians to have skin metastasis.

Key Research Accomplishments, Year 1

- Isolated variables present in eligibility criteria for 85 protocols in PDQ
- Created and implemented structure for storing eligibility criteria and protocols
- Created syntax for representing eligibility criteria, based on modification of Arden syntax
- Implemented parser for extended Arden syntax
- Encoded 85 protocols using structure in XML and Arden syntax
- Implemented simplified patient data model
- Implemented graphical user interface to acquire patient data
- Developed deterministic engine to match patient values against eligibility criteria
- Developed ad-hoc algorithm to rank protocols in reverse order of appropriateness for a particular case
- Implemented graphical user interface to display summarized patient data
- Implemented graphical user interface to display appropriate protocols
- Implemented algorithm to select most informative variables for a given case
- Implemented graphical user interface to display most informative variables
- Started formative evaluation of system's performance
- Started graphical user interface refinement based on oncologist's recommendations
- Redesigned evaluation process
- Obtained approval from IRB to test system with abstracted cases from Brigham and Women's Hospital

Key Research Accomplishments, Year 2

- Updated variables present in eligibility criteria for 85 protocols in PDQ
- Identified changes in protocol status
- Refined structure for representing and storing eligibility criteria and protocols
- Implemented syntax for representing eligibility criteria, allowing all operators from Arden syntax
- Improved parser for Arden syntax
- Refined graphical user interface to acquire patient data, summarize entries and display appropriate protocols
- Improved deterministic engine to match patient values against eligibility criteria
- Informally evaluated algorithm to rank protocols in reverse order of appropriateness for a particular case
- Redesigned evaluation as a clinical trial
- Collected and abstracted real cases from Brigham and Women's Hospital
- Started recruitment for clinical trial

Key Research Accomplishments, Year 3

- Created data model
- Incorporated standard vocabulary
- Redesigned and reimplemented Bayesian networks
- Redesigned graphical user interface
- Created new algorithm for selection and ranking
- Conducted pilot evaluation with two oncologists
- Collected and abstracted real cases from Brigham and Women's Hospital

Key Research Accomplishments, Year 4

- Debugged code from previous version
- Finalized manuscript to be included in the 2002 2001 American Medical Informatics Association Fall Meeting Proceedings.
- Presented final results at the 2001 American Medical Informatics Association Fall meeting in Washington, DC (at student paper competition session and regular session).

Reportable Outcomes, Year 1

Manuscripts

Ohno-Machado L, Boxwala AA, Wang SJ, Mar P. Decision Support for Clinical Trial Eligibility Determination in Breast Cancer. Technical Report TR-199-02, Decision Systems Group.

Abstracts

Ohno-Machado L, Wang SW. Selection of Clinical Trials Using Artificial Intelligence. Abstract for the 1999 Breast Cancer Research Symposium of the Massachusetts Department of Public Health Proceedings

Presentations

Ohno-Machado L, Wang SW. Selection of Clinical Trials Using Artificial Intelligence. Poster presentation at the 1999 Breast Cancer Research Symposium of the Massachusetts Department of Public Health, 4/28/99.

Wang SW, Ohno-Machado L. Selection of Clinical Trials Using Artificial Intelligence. Oral Presentation at the Seminar for the Medical Decision Making Group at the Laboratory for Computer Science, Artificial Intelligence Labs, Department of Electrical Engineering and Computer Science, MIT, 12/8/98.

Informatics such as databases

Database of Encoded Protocols available at http://telmato.bwh.harvard.edu:8000/FACTS/data/

Reportable Outcomes, Year 2

Manuscripts

Ohno-Machado L, Boxwala AA, Wang SJ, Mar P. Decision Support for Clinical Trial Eligibility Determination in Breast Cancer. *Journal of the American Medical Informatics Association* 1999; Suppl 6: 340-4. (best paper award finalist)

Lacson R, Ohno-Machado L. A Comparative Trial of FACTS versus Usual Clinical Practice for Triaging Breast Cancer Patients. Technical Report, Decision Systems Group, Brigham and Women's Hospital and Harvard Medical School, 2000.

Abstracts

Ohno-Machado L, Ogunyemi O, Kogan S. Decision Support for Clinical Trial Eligibility Determination in Breast Cancer. Abstract for the 2000 Breast Cancer Research Symposium of the Massachusetts Department of Public Health Proceedings.

Wang SJ, Ohno-Machado L, Mar P, Boxwala AA, Greenes RA. Enhancing Arden syntax for clinical trial eligibility criteria. Proc *1999 AMIA Annual Fall Symposium,* Washington DC, 1999. Philadelphia: Hanley & Belfus. JAMIA (suppl) 1999: 1188.

Presentations

Ohno-Machado L, Boxwala AA, Wang SJ, Mar P. Decision Support for Clinical Trial Eligibility Determination in Breast Cancer. Presentation at the 1999 AMIA Fall Symposium.

Ohno-Machado L, Ogunyemi O, Kogan S. Decision Support for Clinical Trial Eligibility Determination in Breast Cancer. Poster presented at the 2000 Breast Cancer Research Symposium of the Massachusetts Department of Public Health.

Wang SJ, Ohno-Machado L, Mar P, Boxwala AA, Greenes RA. Enhancing Arden syntax for clinical trial eligibility criteria. Poster presentation at the *1999 AMIA Annual Fall Symposium,* Washington DC, 1999.

Ohno-Machado L, Ogunyemi O, Le H, Greenberg S, Greenes RA. FACTS: Finding Appropriate Clinical Trials. The Internet and the Public's Health: Impact on Individuals, Communities and the World. Poster presentation at theHarvard School of Public Health and Harvard Medical School, May 30-31, 2000.

Reportable Outcomes, Year 3

Manuscripts

Ash, N. New FACTS (*Find Appropriate Clinical TrialS*): A Computer Based Decision Support System for Breast Cancer Patients. Master of Science in Medical Informatics Thesis. Harvard-MIT Division of Health, Sciences and Technology, May 2001.

Abstracts

Ohno-Machado L, Wang S, Greenberg S, Boxwala A. Using the Internet to Find Appropriate Clinical Trials for a Patient: The FACTs project. Proceedings of the Era of Hope, Department of Defense Breast Cancer Research Program Meeting, Atlanta, 2000; 803.

Presentations

Ohno-Machado L, Wang S, Greenberg S, Boxwala A. Using the Internet to Find Appropriate Clinical Trials for a Patient: The FACTs project. Poster presentation at the Era of Hope, Department of Defense Breast Cancer Research Program Meeting, Atlanta, 2000.

Informatics such as databases

Database of Encoded Protocols available at http://dsg.harvard.edu/FACTs/NewFacts/source

Reportable Outcomes, Year 4

Manuscripts

Ash N, Ohno-Machado L, Ogunyemi O, Zeng Q. Finding appropriate clinical trials: evaluating encoded eligibility criteria with incomplete data. Proc AMIA Symp 2001;27-31.

Presentations

Ash N, Ohno-Machado L, Ogunyemi O, Zeng Q. Finding appropriate clinical trials: evaluating encoded eligibility criteria with incomplete data. Proc AMIA Symp 2001;27-31. Presentation in Washington DC for the Student paper competition.

Ash N, Ohno-Machado L, Ogunyemi O, Zeng Q. Finding appropriate clinical trials: evaluating encoded eligibility criteria with incomplete data. Proc AMIA Symp 2001;27-31. Presentation in Washington DC for the regular session.

Personnel

List of salaried personnel who received pay from this grant:

Lucila Ohno-Machado, MD, PhD (PI)

Robert A. Greenes, MD, PhD

Omolola Ogunyemi, PhD

Staal Vinterbo, PhD

Perry Mar, PhD

Darrel Smith, MD, MPH

John Ehresman

Sandra Kogan

Conclusions

We intended to demonstrate the use of a system that could automate the matching of patients to clinical trials, under conditions of uncertainty. Several issues regarding the presentation of the information and the acquisition of conditional probabilities for the Bayesian belief networks that were constructed for this project required further research related to information theory, human-computer interaction, and reasoning with uncertainty. We have accomplished the overall tasks of the system towards the construction of a prototype automated system to automate patient eligibility match to suggest appropriate protocols for a specific patients [21]. Earlier prototypes were redesigned given user's feedback. We have implemented engines that deal with uncertain items and infer appropriate values. We have evaluated the system and compared its performance with that of two oncologists using data from the electronic medical record at Brigham and Women's Hospital. We have concluded that the addition of reasoning under uncertainty can be beneficial but the trade-offs between model complexity and manageability need to be taken into account in such systems.

References

1. Ohno-Machado L, Boxwala AA, Wang SJ, Mar P. Decision Support for Clinical Trial Eligibility Determination in Breast Cancer. *Journal of the American Medical Informatics Association* 1999; Suppl 6: 340-4.
2. CancerNet - search for clinical trials (PDQ - Physician Data Query) - A National Cancer Institute service, in http://cancernet.nci.nih.gov/trialsrch.shtml.
3. Silva J and Wittes R, Role of clinical trials informatics in the NCI's cancer informatics infrastructure. Proc AMIA Symp, 1999:950-4.
4. Together - the control center for accelerating e-solutions, v. 3.2, TogetherSoft, Raleigh, North Carolina.
5. Lindberg D, The Unified Medical Language System (UMLS) of the National Library of Medicine. J Am Med Rec Assoc, 1990;61:40-2.
6. Ogunyemi O, The Guideline Expression Language (GEL) User's guide, Technical Report, DSG-TR-2000-001, 2000, Decision Systems Group, Boston, MA.
7. Hripcsak G, Writing Arden Syntax Medical Logic Modules. Comput Biol Med, 1994;24:331-63.
8. Shortliffe EH, Barnett GO, Cimino JJ, et al., Collaborative medical informatics research using the Internet and the World Wide Web. Proc AMIA Annu Fall Symp, 1996:125-9.
9. Ohno-Machado L, Gennari JH, Murphy SN, et al., The guideline interchange format: a model for representing guidelines. J Am Med Inform Assoc, 1998;5:357-72.
10. Peleg M, Boxwala AA, Ogunyemi O, et al., GLIF3: the evolution of a guideline representation format. Proc AMIA Symp, 2000:645-9.
11. Pearl J, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. 1988, San Mateo,CA: Morgan Kaufmann.
12. Tu SW, Kemper CA, Lane NM, et al., A methodology for determining patients' eligibility for clinical trials. Methods Inf Med, 1993;32:317-25.
13. Ohno-Machado L, Parra E, Henry SB, et al., AIDS2: a decision-support tool for decreasing physicians' uncertainty regarding patient eligibility for HIV treatment protocols. Proc Annu Symp Comput Appl Med Care, 1993:429-33.
14. Papaconstantinou C, Theocharous G, and Mahadevan S, An expert system for assigning patients into clinical trials based on Bayesian networks. J Med Syst, 1998;22:189-202.
15. Cozman FG, JavaBayes - a system that handles Bayesian networks implemented in Java, v. 0.346.
16. Ries LAG WP, Miller DS, Howe HL, Weir HK, Rosenberg HM, Vernon SW, Cronin K, Edwards BK, The Annual Report to the Nation on the Status of Cancer, 1973-1997, With a Special Section on Colorectal Cancer. Cancer, 2000;88:2398-2424.

17. Neapolitan R, Probabilistic reasoning in expert systems: Theory and algorithms. 1990, New York: Wiley, John & Sons, Inc.

18. Daniel Ww, Biostatistics: a foundation for analysis in the health sciences. Wiley series in probability and mathematical statistics - applied. 1995, New York: Wiley, John & Sons, Inc.

19. Altman D, Practical Statistics for Medical Research. 1991: CRC press, LLC. 403-409.

20. Analyse-it! Statistical analysis add-in software for Microsoft Excel, v. 1.61, Analyze-it Software LTD, Leeds, UK.

21. Ash N, Ohno-Machado L, Ogunyemi O, Zeng Q. Finding appropriate clinical trials: evaluating encoded eligibility criteria with incomplete data. Proc AMIA Symp 2001;27-31.

List of Appendices

Decision Support for Clinical Trial Eligibility Determination in Breast Cancer

Lucila Ohno-Machado, MD, PhD, Samuel J. Wang, MD, PhD,
Perry Mar, Aziz A. Boxwala, MBBS, PhD
Decision Systems Group and Division of Health Sciences and Technology
Harvard Medical School and Massachusetts Institute of Technology, Boston, MA

ABSTRACT

We have developed a system for clinical trial eligibility determination where patients or primary care providers can enter clinical information about a patient and obtain a ranked list of clinical trials for which the patient is likely to be eligible. We used clinical trial eligibility information from the National Cancer Institute's Physician Data Query (PDQ) database. We translated each free-text eligibility criterion into a machine executable statement using a derivation of the Arden Syntax. Clinical trial protocols were then structured as collections of these eligibility criteria using XML. The application compares the entered patient information against each of the eligibility criteria and returns a numerical score. Results are displayed in order of likelihood of match. We have tested our system using all phase II and III clinical trials for treatment of metastatic breast cancer found in the PDQ database. Preliminary results are encouraging.

INTRODUCTION

Historically, accrual of patients for clinical trials has not been very successful, particularly for certain clinical domains. Studies demonstrate that just a small percentage of eligible patients (3 to 10%) are actually enrolled in such trials [1,2]. The low accrual rates are attributed to: (1) physician factors such as lack of knowledge about clinical trials, (2) patient factors such as lack of patient-oriented information regarding trials, (3) organizational barriers, and (4) health care system obstacles. If clinical trial information can be made more accessible to patients and their primary care providers (PCPs), we believe that clinical trial accrual rates can improve.

The increasing participation of patients in decisions regarding their own health has created a demand for health information resources oriented towards the patient and PCP, rather than the specialist [5]. A few systems have been previously designed to help with the determination of clinical trial eligibility. Tu et al. developed systems for this purpose, described in [6]. Ohno-Machado et al. previously developed a system that could reason under conditions of uncertainty [7]. However, these systems have focused on helping investigators identify eligible patients for a specific clinical trial. In contrast to these systems, the purpose of our system is to enable PCPs and patients to identify the best trials for a specific patient.

MATERIALS AND METHODS

Data. We used the National Cancer Institute's Physician Data Query (PDQ) database [8] as the source of information for clinical trials. The clinical trial summaries in the PDQ database contain free-text lists of eligibility criteria organized by patient characteristics (e.g., age, menopausal status); disease characteristics (e.g., histology, metastases); and prior and concurrent therapy. For the preliminary phase of this study, we selected from the PDQ database all Phase II and Phase III trials for the treatment of metastatic or recurrent breast cancer. Breast cancer was chosen because this is the oncology domain that contains the largest number of clinical trials. We chose advanced stage cancer because we hypothesized that these patients would be more interested in seeking participation in clinical trials after exhausting traditional treatment venues. We decided to limit our initial set to Phase II and Phase III trials since these studies are further developed, and typically involve several patients. We found a total of 85 clinical trials in the PDQ database (as of July 1998) that fit these parameters.

Clinical Trial Eligibility Database. Each clinical trial summary was encoded into a structured format. The encoded summary was stored in an XML document (Figure 1). This document contains elements describing identifying information about the clinical trial (name of trial, protocol number) and a collection of criteria elements. Each criterion element contains the original narrative text description from PDQ and the criterion encoded in a computable expression. The criterion is encoded in a modified version of the grammar used for specifying logic statements in the Arden Syntax [9]. Modifications had to be made to the Arden Syntax specification in order to accommodate a data model that contains hierarchical term relationships and compound data-types. (Details and discussion of our modifications to the Arden Syntax are presented elsewhere [10].) The resulting extended syntax for conditional expressions is also being incorporated into proposed extensions to GLIF, a clinical guideline interchange format developed by The InterMed Collaboratory [11].

```
<PROTOCOL ID="09251">
 <NAME> Phase II Randomized Study of
Cyclophosphamide/Methotrexate/Fluorouracil (CMF)
Vs Mitoxantrone in Elderly Patients with Advanced Breast
Cancer
 </NAME>

<!--Disease Characteristics-->
<CRITERION>
  No CNS metastases
  <SPEC>
    [metastases_locations where it is a "CNS"] == []
  </SPEC>
</CRITERION>

<!--Patient Characteristics-->
<CRITERION>
 Over 70
  <SPEC>
    age &gt; 70
  </SPEC>
</CRITERION>

<CRITERION>
 Postmenopausal
   <SPEC>
     menopausal_status == 'postmenopausal'
   </SPEC>
</CRITERION>

<!--Hematopoietic:-->
<CRITERION>
 WBC at least 3,000
   <SPEC>
    WBC &gt;= 3000
   </SPEC>
 </CRITERION>
```

Figure 1. Excerpt of clinical trial protocol structured in XML format.

The translation of the original free-text criterion descriptions from PDQ into a machine-interpretable representation was largely a manual process performed by informatics fellows and faculty in our laboratory. We used text parsing tools such as Perl scripts to automate portions of this process. We established a uniform basis for encoding criteria. For example, a certain clinical trial summary may have specified "estrogen receptor negative," and another may have specified "ER (-)." These refer to the same eligibility criterion and are encoded using the same expression ("estrogen_receptor == negative").

```
<!-- Patient Characteristics -->

<VARIABLE NAME='age' TYPE='number' CUI='C0001779'>
</VARIABLE>

<VARIABLE NAME='birthdate' TYPE='date' CUI='C0421451'>
</VARIABLE>

<VARIABLE NAME='gender' TYPE='enum' CUI='C0079399'>
Gender of patient
  <VALUE CUI='C0024554'>male</VALUE>
  <VALUE CUI='C0015780'>female</VALUE>
</VARIABLE>

<VARIABLE NAME='menopausal_status' TYPE='enum'
CUI='C0025320'>
Menopausal status of patient.
  <VALUE CUI='C0279752'>premenopausal</VALUE>
  <VALUE CUI='C0279753'>postmenopausal</VALUE>
</VARIABLE>
```

Figure 2. Excerpts from data dictionary containing definitions of clinical concepts used in the eligibility criteria.

In order to adequately model eligibility criteria, we found it necessary to create a data model that was sophisticated enough to accommodate hierarchical relationships among clinical concepts, sub-attributes of concepts, and temporal relationships among concepts. The concepts used in the eligibility criteria were defined in a data dictionary (also an XML document) (Figure 2), and mapped to concepts in the UMLS Metathesaurus [12]. We analyzed all the encoded criteria to assess which concepts occurred most frequently and were also relatively easy for the patient or PCP to obtain. This information was taken into consideration to construct web-based entry forms, shown in Figure 3.

Clinical Trial Ranking. Upon entry of patient data, the application produces a ranked list of clinical trials that the patient is eligible for. The ranking algorithm is tolerant of missing data. All criteria are considered as having equal weight (importance) when used in protocol ranking. The algorithm sequentially processes all the criteria in all the clinical trials. The algorithm first rules out all clinical trials for which at least one eligibility criterion was not met. For the remaining clinical trials, the ones that have fewest unknown criteria are placed higher on the list. Resulting trials are displayed with links to the original PDQ clinical trial summaries (Figure 4). The search can be refined with data entered in dynamically created forms (Figure 5). For each clinical trial, we also provide a summary of which criteria have been met and which still need to be evaluated (Figure 6).

Application. We are developing two versions of the application: one for the primary care provider and one for the patient. The version for the patient will provide a simplified user interface and will only request data that a patient would be expected to know. The application runs on the Microsoft Windows platform. HTML pages are dynamically generated on the server using Microsoft's Active Server Pages (ASP). The application logic was written in Visual C++ and wrapped as an ActiveX object that is invoked by ASP.

RESULTS

A total of 2188 criteria in the set of 85 clinical trials were chosen for this study. In this set, the least, most, and median number of criteria in a protocol were 6, 45, and 25 respectively. To date, we have encoded about 50% of the criteria in these clinical trials. We are first encoding frequently occurring criteria and those that are readily accommodated by the criteria representation syntax. (See [10] for details on difficulties encountered in encoding the eligibility criteria.) Figures 3 to 6 show an example of the PCP version of the application for a sample breast cancer patient: a premenopausal, 55 year-old woman with stage IV breast cancer with metastases to liver and bone, previous mastectomy, chemotherapy and

radiotherapy. This patient also suffers from coronary artery disease and diabetes mellitus. Figure 3 shows the initial data input form in which the PCP has entered some clinical information about the patient. Using this information, the program returns a preliminary list of trials. This list is ranked, with the most likely matches at the top (Figure 4).

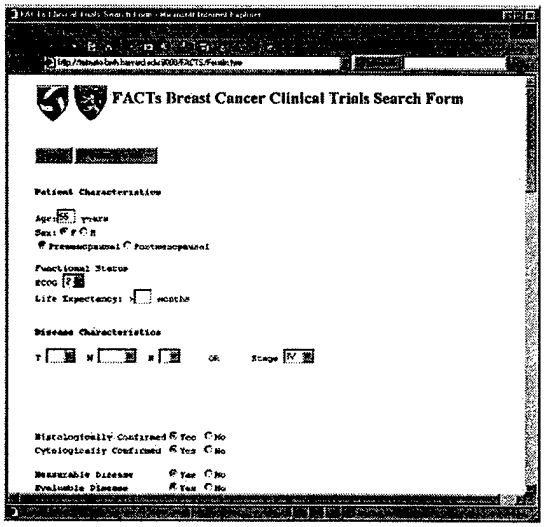


Figure 3. The initial entry form requests items that are most frequent and easiest to obtain.

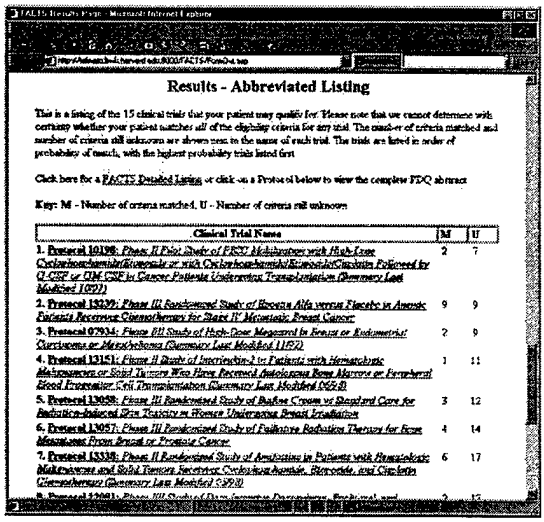


Figure 4. Results page showing a ranked list of clinical trials.

If the list is long, the application offers the PCP an opportunity to fill in additional patient information to narrow the search. The program dynamically constructs the secondary input form to request the information that would be more likely to narrow the number of clinical trials (Figure 5). Again, the PCP fills in as much additional information as he or she can. This process can be repeated as many times as desired until either the resulting list is short enough, or there is no additional information required or available.

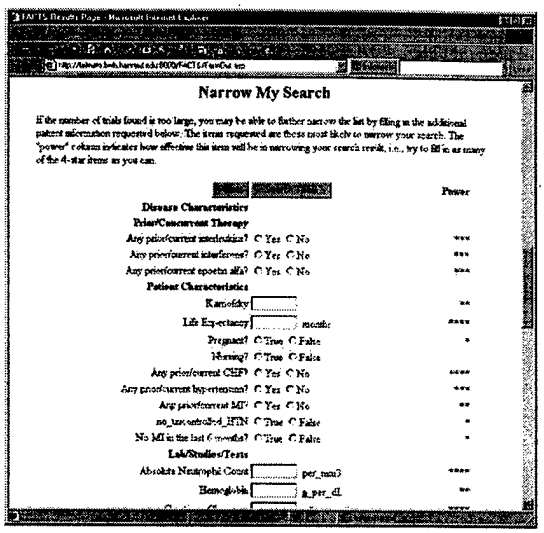


Figure 5. Secondary entry forms are created dynamically and request information that will be most useful in narrowing the search.

The final list is presented in order of likelihood of match. In this example, the system narrowed the list to 15 trials that the patient is potentially eligible for. A summary of all the entered information is provided. Detailed information about these clinical trials (Figure 6) can be displayed, along with a list of the criteria still to be checked.

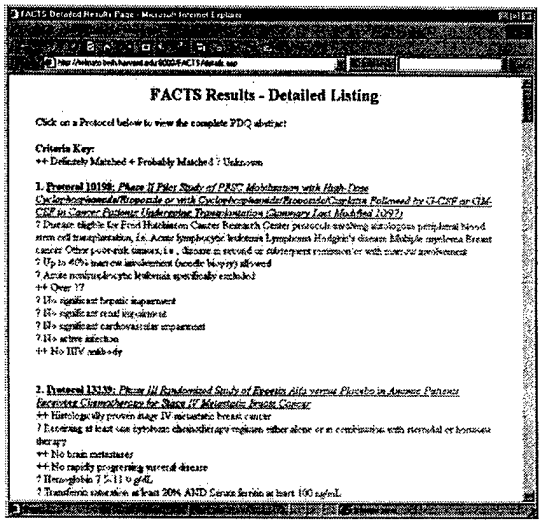


Figure 6. Detailed information about remaining trials is displayed.

Discussion and Future Directions

The current ranking algorithm makes two simplifying assumptions: (1) all criteria have equal importance and equal probability of being met if their values are

unknown, and (2) all criteria are independent. Regarding the first assumption, a more accurate approach would be to assign a weight to each criterion or data item, and then use these weights to compute the ranking. We may be able to obtain these weights by asking domain experts, from the literature, or by analysis of large patient data sets. Tu [6] has proposed that some criteria variables are mutable over time (e.g., age) or controllable (e.g., stop current chemotherapy), and therefore might bear less weight in ruling-out or ranking one clinical trial against others. We have not decomposed criteria into "atomic" parts, each containing just one variable, hence this approach has not been yet tested.

The other simplifying assumption, criteria (and data item) independence, also introduces inaccuracies in ranking. For example, a clinical trial may specify two separate data items for the liver function tests, AST and ALT: "AST < 2 times normal" and "ALT < 2 times normal." These criteria are currently considered independent, when in fact a better approximation would be to consider them just *conditionally* independent given a certain liver disease. For example, if AST is high, there is an increased probability that ALT is high because the disease that causes the former to increase is also likely to cause the latter to do so. The independence assumption causes some criteria to be unfairly "counted twice." A more accurate approach would be to identify dependencies among the data items and adjust the scoring accordingly. In this version of the application, we considered all criteria to be Boolean (i.e., "true" or "false"), and have not further characterized their nature.

The current clinical trial selection algorithm is deterministic. We have not attempted to deal with uncertainty using probabilities in this prototype. A global model to infer the value of missing values for common criteria and specification of criteria dependencies will be built using expert knowledge. This model will be based on a belief network, the structure and probabilities of which will be extracted by interviews with specialists, analysis of literature, or "learned" from clinical databases. A future version of this system will take into account "proxies" for certain criteria (e.g., known renal disease as a proxy for laboratory values that measure renal function, or "severity of cancer" as a proxy for staging). The probabilities of eligibility will be determined by inferencing values for required data from the proxies.

Other prototype applications have been built with the assumption that certain medical domains may require very few eligibility criteria to reasonably eliminate a large percentage of the candidate trials for a given patient [13]. In contrast, our approach has been to attempt to encode as many criteria as we reasonably can in an attempt to arrive at a more accurate list of potentially matching clinical trials. However, it is difficult to algorithmically determine eligibility with 100% accuracy because of the clinical judgement that is necessary for evaluating several of these criteria. Our objective is to narrow and rank the list of matching trials, as much as possible, before turning the list over to a specialist for final determination of eligibility. Encoding complex criteria is a time-consuming effort. Although we have developed some automated parsing tools to facilitate this task, it remains a largely manual process. We predict that our application will perform better as we encode more criteria. However, an open question that deserves further study is how much encoding is "enough," i.e., at what point is it not cost-beneficial to encode more complex criteria. Since software applications cannot determine clinical trial eligibility with 100% certainty, it may not be worth the extra effort to encode very complex criteria.

The criteria encoded for this study were taken from clinical trial summaries from PDQ. These summaries are abstracted from the original protocol documents and may lose some fidelity in the process. Our encoding is only as good as the translated text descriptions. For improving accuracy, an alternative approach would be to go directly to the original full research clinical trial descriptions to obtain the eligibility criteria. The future development and routine use of computer-based protocol authoring tools may reduce these problems.

Currently, we have not taken into account patient preferences in ranking the clinical trials, such as modality of treatment, potential toxicity, potential for cure, and geographic constraints. The system currently ranks trials solely based on the likelihood that the patient will satisfy the eligibility requirements. It is a very different question to ask what types of trials a patient may prefer. While eligibility criteria are obviously a firm prerequisite to enrollment, in cases with incomplete information, there may be some benefit to introducing patient preferences even before eligibility has been completely determined. This could help narrow the list more quickly so as not to waste the patient's or clinician's time in reviewing eligibility requirements for trials that the patient would never consider enrolling in.

We plan to automatically retrieve some of the required patient data from the clinical information system at our institution in order to ease the data entry burden on the user. The user will only need to provide information not available in the clinical system. For the institutional version, we will link the

eligibility component to other tools that automate the enrollment process, such as display of informed consent forms, and detailed explanation of the clinical trials. A more general version of the application will be available on the WWW. In addition to UMLS, we also plan to map the concepts used in our system to the Common Data Elements (CDE) that are being developed under the supervision of the informatics group at the National Cancer Institute [14]. Mapping to the CDE will make the system more robust for national scale use. The open architecture and facility to add customized dictionaries will also make it easy to adapt the system for integration to electronic medical record systems of different institutions.

This initial version of the application has been designed for use by PCPs. For the patient version, we intend to customize the user interface according to different levels of user sophistication. The user interface will be designed in consultation with patient advocacy groups, health educators, and PCPs. Reduction and simplification of data items to be entered is necessary. We will utilize a decision analytic approach to determine the data items needed.

CONCLUSIONS

We have developed a WWW-based decision support system to help patients and providers determine the patient's eligibility for certain clinical trials. The system currently contains all Phase II and III treatment clinical trials for metastatic breast cancer from the NCI's PDQ database. It rules out trials that the patients are not eligible for and ranks the remaining trials according to how many criteria still need to be checked to determine eligibility. This initial prototype system has helped us identify relevant issues in machine-readable criteria representation, user interface design, and clinical trial ranking under uncertainty. Preliminary testing of the system with a few clinical cases has been promising. A formal evaluation of usability and reliability is underway. Future versions of this application will include a belief network that will allow the system to impute missing data values and reason under conditions of uncertainty.

ACKNOWLEDGMENTS

This project was funded by contract 34078PP1024 from the Massachusetts Department of Public Health and grant DAMD17-98-1-8093 from the Department of the Army. The contents of this article do not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Dr. Wang was funded by NLM grant 2 T15 LM07092. We would like to thank Jeremy Theal, Dr. Jeff Huhn and Dr. Ross Martin for helping to encode the eligibility criteria. We also thank Dr. Ursula Matulonis, Dr. Craig Bunnel, and Dr. Darrel Smith for sharing their expertise in breast cancer. Mr. John Ehresman implemented parts of this work. Prof. Robert Greenes provided valuable suggestions to this project.

REFERENCES

[1] Mansour EG. Barriers to clinical trials. Part III: Knowledge and attitudes of health care providers. *Cancer*, 1994, 74(9 Suppl):2672-5.

[2] Winn RJ. Obstacles to the accrual of patients to clinical trials in the community setting. *Seminars in Oncology*, 1994, 21(4 Suppl 7):112-7.

[3] Taylor KM; Margolese RG; Soskolne CL. Physicians' reasons for not entering eligible patients in a randomized clinical trial of surgery for breast cancer. *New England Journal of Medicine*, 1984, 310(21):1363-7.

[4] Klabunde C; Kaluzny A; Ford L. Community Clinical Oncology Program participation in the Breast Cancer Prevention Trial: factors affecting accrual. *Cancer Epidemiology, Biomarkers and Prevention*, 1995, 4(7):783-9.

[5] Sweeney MA; Skiba D. Combining telecommunications and interactive multimedia health information on the electronic superhighway. *Medinfo*, 1995, 8 Pt 2:1524-7.

[6] Tu SW; Kemper CA; Lane NM; Carlson RW; Musen MA. A methodology for determining patients' eligibility for clinical trials. *Methods of Information in Medicine*, 1993, 32(4):317-25.

[7] Ohno-Machado L, Parra E, Henry SB, Tu SW, Musen MA. AIDS2: A decision-support tool for decreasing physicians' uncertainty regarding patients eligibility for HIV treatment protocols. *Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care,* 429-33, 1993.

[8] NCI. http://cancernet.nci.nih.gov/.

[9] American Society for Testing and Materials. E 1460 Standard Specification for Defining And Sharing Modular Health Knowledge Bases (Arden Syntax for Medical Logic Modules). ASTM Standards, v 14.01. Philadelphia: ASTM, 1992; 539-87.

[10] Wang SJ, Ohno-Machado L, Mar P. Representing Criteria in Guidelines and Clinical Trial Protocols: Common Needs and Solutions. *Decision Systems Group, Brigham and Women's Hospital, Technical Report* 1999-04.

[11] Ohno-Machado L, Gennari JH, Murphy SN, et al. The guideline interchange format: a model for representing guidelines. *J Am Med Inform Assoc*. 1998;5:357-72.

[12] Humphreys BL, Lindberg DA. The UMLS project: making the conceptual connection between users and the information they need. Bull Med Libr Assoc 1993;81(2):170-7.

[13] Gennari J. personal communication, 1998.

[14] NCI Cancer Informatics Infrastructure Home Page. http://hiip-wkstn.hpc.org/.

Technical Report
DSG-TR-2000-001

The Guideline Expression Language (GEL) User's Guide

Omolola Ogunyemi

oogunyem@dsg.harvard.edu

20 November 2000

Decision Systems Group
Brigham and Women's Hospital
Harvard Medical School
75 Francis Street, Boston, MA 02115

The Guideline Expression Language (GEL) User's Guide

Types supported by GEL are listed below and expressions involving constants of these types are provided as examples of how to write valid expressions in GEL. A variable in GEL can be assigned a value of any one of the types described below:

Number (real numbers)
String
Extended Boolean (true, false, unknown)
Absolute Date and Time
Duration
List
Numeric Interval
Duration Interval
Absolute Date and Time Interval

Number

Operations supported on numbers include comparisons, addition, subtraction, multiplication, division, exponentiation, unary plus, and unary minus. A number in GEL is a floating point/real number by default. Use of unsupported operators with numerical values is an error (causes a type mismatch exception to be raised).

Unary operators:

\+
Description: unary plus operator
Sample expression: (+3)
Returns: 3
Note: the parentheses are required

\-
Description: unary minus operator
Sample expression: (-50)
Returns: -50
Note: the parentheses are required

is number
Description: checks type of argument and returns true if it is a number
Sample expression: is number 225
Returns: true
Sample expression: is number "hey"
Returns: false

Binary operators:

\+
Description: addition operator
Sample expression: 2 + 3
Returns: 5

\-
Description: subtraction operator
Sample expression: 2 - 3
Returns: -1

*
Description: multiplication operator
Sample expression: 50 * (-3)
Returns: -150

/

Description:	division operator
Sample expression:	180 / 6
Returns:	30
Sample expression:	22 / 7
Returns:	3.142857142857143

^ *or* **

Description:	exponentiation operator
Sample expression:	2 ^ 5
Returns:	32
Sample expression:	3 ** 6
Returns:	729
Sample expression:	2 ^ (-4)
Returns:	0.0625

<

Description:	less than operator
Sample expression:	5 < 4
Returns:	false

>

Description:	greater than operator
Sample expression:	(-9) > (-18)
Returns:	true

<=

Description:	less than or equal to operator
Sample expression:	51 <= 51
Returns:	true

>=

Description:	greater than or equal to operator
Sample expression:	200 >= 165
Returns:	false

= *or* ==

Description:	equality operator
Sample expression:	20 == 12
Returns:	false
Sample expression:	1 = 1
Returns:	true

!= *or* <>

Description:	inequality operator
Sample expression:	20 <> 12
Returns:	true
Sample expression:	1 != 1
Returns:	false

Ternary operators:

is within ... to ...

Description:	checks that first argument is in the inclusive range defined by the second and third arguments
Sample expression:	5 is within 4 to 5
Returns:	true
Sample expression:	10 is within 2 to 9
Returns:	false

String

Operations supported on strings include concatenation and lexicographic comparisons. Use of unsupported operators with string values is an error (causes a type mismatch exception to be raised).

Unary operators:

is string
Description: checks type of argument and returns true if it is a string
Sample expression: is string 225
Returns: false
Sample expression: is string "hey"
Returns: true

Binary operators:

|| *or* concat
Description: concatenation operator
Sample expression: "hello " || "world"
Returns: "hello world"
Sample expression: "thirty-" concat "four"
Returns: "thirty-four"

<
Description: less than operator (checks whether the 1st argument lexicographically precedes the 2nd argument)
Sample expression: "a" < "aa"
Returns: true
Sample expression: "d" < "b"
Returns: false

>
Description: greater than operator (checks whether the 1st argument lexicographically follows the 2nd argument)
Sample expression: "yy" > "ab"
Returns: true

<=
Description: less than or equal to operator (checks whether the 1st arg. lexicographically precedes or equals the 2nd)
Sample expression: "cd" <= "cd"
Returns: true

>=
Description: greater than or equal to operator (checks whether the 1st arg. lexicographically follows or equals the 2nd)
Sample expression: "zed" >= "zee"
Returns: false

= *or* ==
Description: equality operator
Sample expression: "why" == "not"
Returns: false

!= *or* <>
Description: inequality operator
Sample expression: "why" <> "not"
Returns: true

Ternary operators:

is within ... to ...
Description: checks that first argument is in the inclusive range defined by the second and third arguments
Sample expression: "aa" is within "a" to "b"
Returns: true
Sample expression: "c" is within "cc" to "ea"
Returns: false

Extended Boolean

Extended booleans in the expression language describe a 3-valued logic (true, false, and unknown). Operations on extended booleans include logical ands, ors, xors, etc. Use of unsupported operators with extended boolean values is an error (causes a type mismatch exception to be raised).

Unary operators:

is boolean
Description: checks type of argument and returns true if it is an extended boolean
Sample expression: is boolean unknown
Returns: true
Sample expression: is boolean 0
Returns: false

is unknown
Sample expression: is unknown true
Returns: false
Sample expression: is unknown false
Returns: false
Sample expression: is unknown unknown
Returns: true

not *or* !
Description: logical not
Sample expression: not true
Returns: false
Sample expression: ! false
Returns: true
Sample expression: not unknown
Returns: unknown

any of
Description: returns true if any of the logical expressions in its argument evaluates to true. Expects a comma separated "list" of logical expressions as its argument.
Sample expression: any of (3>4, 67 < 99, true == true, true xor false)
Note, equivalent to: any of (false, true, true, true)
Returns: true

all of
Description: returns true if all of the logical expressions in its argument evaluate to true. Expects a comma separated "list" of logical expressions as its argument.
Sample expression: all of (3>4, 67 < 99, true == true, true xor false)
Note, equivalent to: all of (false, true, true, true)
Returns: false

Binary operators:

= *or* ==
Description: equality operator
Sample expression: true == unknown
Returns: false

!= *or* <>
Description: inequality operator
Sample expression: false != unknown
Returns: true

and *or* &
Description: logical and

Sample expression:	true and true
Returns:	true
Sample expression:	true and false
Returns:	false
Sample expression:	true and unknown
Returns:	unknown
Sample expression:	false & false
Returns:	false
Sample expression:	false & unknown
Returns:	false
Sample expression:	unknown & unknown
Returns:	unknown

or *or* |

Description:	logical or
Sample expression:	true or true
Returns:	true
Sample expression:	true or false
Returns:	true
Sample expression:	true or unknown
Returns:	true
Sample expression:	false | false
Returns:	false
Sample expression:	false | unknown
Returns:	unknown
Sample expression:	unknown | unknown
Returns:	unknown

xor *or* *|

Description:	exclusive or
Sample expression:	true xor true
Returns:	false
Sample expression:	true xor false
Returns:	true
Sample expression:	true xor unknown
Returns:	unknown
Sample expression:	false *| false
Returns:	false
Sample expression:	false *| unknown
Returns:	unknown
Sample expression:	unknown *| unknown
Returns:	unknown

The following binary operator expects a number followed by a comma-separated list of logical expressions:

at least ... of ...

Description:	returns true if the number of logical expressions in its right argument that evaluate to true equal or exceed its numeric argument.
Sample expression:	at least 2 of (3>4, 67 < 99, true == true, true xor false) Note, equivalent to: at least 2 of (false, true, true, true)
Returns:	true
Sample expression:	at least 5 of (3>4, 67 < 99, true == true, true xor false) Note, equivalent to: at least 5 of (false, true, true, true)
Returns:	false

Absolute Date and Time

Absolute dates and times and operations on them are defined with respect to a Gregorian calendar. Operations on absolute dates and times include comparisons, subtraction, etc. Use of unsupported operators with absolute date and time values is an error (causes a type mismatch exception to be raised). An absolute date and time value that does not end in a Z for universal coordinated time (UTC) or in

a +/- hh:mm offset is assumed to be defined in local time. Note that the expression **now** yields the current time on the particular system running an interpreter for GEL.

Unary operators:

is time
Description: checks type of argument and returns true if it is an absolute date and time
Sample expression: is time 1999-03-04T03:30:45.742-03:00
Returns: true
Sample expression: is time 2000-09-12
Returns: true
Sample expression: is time **now**
Returns: true
Sample expression: is time 23
Returns: false

extract date
Description: extracts the date portion of the argument and returns it as an absolute date and time in local time
Sample expression: extract date 1998-03-04T03:30:45.742+05:30
Returns: 1998-03-04
Sample expression: extract date **now** (assuming **now** is 2000-10-03T17:59:10.240-04:00)
Returns: 2000-10-03

extract year
Description: extracts the year portion of an absolute date and time
Sample expression: extract year 1998-03-04T03:30:45.742-03:00
Returns: 1998

extract month
Description: extracts the month portion of an absolute date and time
Sample expression: extract month 2001-11-05
Returns: 11

extract day
Description: extracts the day of the month from an absolute date and time
Sample expression: extract day 1950-12-25
Returns: 25

extract hour
Description: extracts the hour of the day from an absolute date and time
Sample expression: extract hour 1960-10-01T03:04:30
Returns: 3

extract minute
Description: extracts the number of minutes past the hour from an absolute date and time
Sample expression: extract minute 1960-10-01T03:04:30
Returns: 4

extract second
Description: extracts the number of seconds past the hour from an absolute date and time
Sample expression: extract second 1960-10-01T03:04:30
Returns: 30

Binary operators:

-
Description: subtract one absolute date and time from another to produce a duration in seconds
Sample expression: 2000-03-01T00:00:00 - 2000-02-01T00:00:00
Returns: 2505600 seconds

occurs at

Description: checks that first argument and the second argument are equal
Sample expression: 2000-03-10T05:04:03 occurs at 2000-03-10T12:55:43
Returns: false
Sample expression: 2000-03-10T00:00:00 occurs at 2000-03-10T23:59:59
Returns: false
Sample expression: 2000-03-10T05:04:03 occurs at 2000-03-10T05:04:03
Returns: true

is within same day as
Description: checks that first argument and the second argument occur on the same day (a new day begins at midnight)
Sample expression: 2000-03-10T05:04:03 is within same day as 2000-03-10T12:55:43
Returns: true
Sample expression: 2000-03-10T00:00:00 is within same day as 2000-03-10T23:59:59
Returns: true
Sample expression: 2001-03-10T05:04:03 is within same day as 2000-03-10T12:55:43
Returns: false

is before
Description: determines whether one date occurs before another
Sample expression: 2000-03-01T00:00:00 is before 2000-02-01T00:00:00
Returns: false

is after
Description: determines whether one date occurs before another
Sample expression: 2000-03-01T00:00:00 is after 2000-02-01T00:00:00
Returns: true

<
Description: less than operator (equivalent to is before)

>
Description: greater than operator (equivalent to is after)

<=
Description: less than or equal to operator

>=
Description: greater than or equal to operator

= *or* ==
Description: equality operator (same as occurs at)
Sample expression: 2010-03-01T00:00:00 == 2009-03-01T00:00:00
Returns: false

!= *or* <>
Description: inequality operator
Sample expression: 2010-03-01T00:00:00 != 2009-03-01T00:00:00
Returns: true

The following binary operators expect a time followed by a duration:

is within past
Description: checks that first argument is within the duration specified by **now** minus the second argument to **now**
Sample expression: 2000-10-02T00:00:00 is within past 2 days (assuming that **now** is 2000-10-04T19:04:18.650-04:00)
Returns: false
Note: this operator calculates past two 2 days as **48 hours before the present time**
If two days prior is meant to start at midnight, other expressions could be substituted such as:
(2000-10-02T00:00:00 >= extract date (2 days ago)) and (2000-10-02T00:00:00 <= extract date now)
Sample expression: 2000-10-02T23:30:00 is within past 2 days (assuming that **now** is 2000-10-04T19:04:18.650-04:00)
Returns: true

-

Description:	Subtracts a duration from an absolute date and time
Sample expression:	now – 3 days (assuming now is 2000-10-20T15:03:38.419-04:00)
Returns:	2000-10-17T15:03:38.419-04:00
Sample expression:	1998-01-31 - 28 days
Returns:	1998-01-03T00:00:00-05:00

The following binary operators expect a time and a duration as arguments (in no particular order):

+

Description:	Adds a duration to an absolute date and time
Sample expression:	1995-03-04 + 720 days
Returns:	1997-02-21T00:00:00-05:00
Sample expression:	5 hours + 1999-03-04T05:00:00
Returns:	1999-03-04T10:00:00-05:00

Ternary operators:

... is within ... to ...

Description:	checks that first argument is in the inclusive range defined by the second and third arguments
Sample expression:	2000-03-10T05:04:03 is within 2000-03-10T05:04:03 to 2000-05-10T05:04:03
Returns:	true

The following ternary operators expect as arguments a time followed by a duration followed by a time:

... is within ... preceding ...

Description:	checks that first argument is in the inclusive range defined by the third argument minus the second argument to the third argument
Sample expression:	2000-03-10T05:04:03 is within 4 months preceding 2000-05-10T05:04:03
Returns:	true

... is within ... following ...

Description:	checks that first argument is in the inclusive range defined by the third argument to the third argument plus the second argument
Sample expression:	2000-10-03T06:45:23 is within 5 days following 2000-10-01T00:55:46
Returns:	true

... is within ... surrounding ...

Description:	checks that first argument is in the inclusive range defined by the third argument minus the second argument to the third argument plus the second argument
Sample expression:	2000-09-29T17:20:01 is within 5 days surrounding 2000-10-01T00:55:46
Returns:	true
Sample expression:	2000-10-05T00:00:00 is within 5 days surrounding 2000-10-01T00:55:46
Returns:	true
Sample expression:	2000-10-06T19:05:40 is within 5 days surrounding 2000-10-01T00:55:46
Returns:	false
Sample expression:	(extract date 2000-10-06T19:05:40) is within 5 days surrounding (extract date 2000-10-01T00:55:46)
Returns:	true

Duration

Operations supported on durations include comparisons, addition, subtraction, multiplication, and division. Use of unsupported operators with duration values is an error (causes a type mismatch exception to be raised. Note that because of the fuzziness associated with certain durations (is 1 year 365 or 366 days? Is 1 month 28, 29, 30, or 31 days?), defaults are used for the number of days in a year (1 year = 365 days in our model), and the number of days in a month (1 month = 31 days in our model). This means that certain operators would return results that differ from the expected. For example the query 1 year == 12 months would return false because 365 days is not equal to 372 (12*31) days.

Ultimately, the best approach to evaluating such fuzzy or vague comparisons might be to apply appropriate methods for handling uncertainty from the Artificial Intelligence literature on uncertainty, or to disallow precise calculations from being made from such imprecise expressions.

Unary Operators

is duration
Description: checks type of argument and returns true if it is a duration
Sample expression: is duration 3 years
Returns: true
Sample expression: is duration 5 months
Returns: true
Sample expression: is duration 20 hours
Returns: true
Sample expression: is duration 23
Returns: false

ago
Description: computes an absolute date and time equivalent to the current time (**now**) minus a duration
Sample expression: 2 days ago (assuming **now** is 2000-10-03T18:19:06.270-04:00)
Returns: 2000-10-01T18:19:06.270-04:00

from now
Description: computes an absolute date and time equivalent to the current time (**now**) plus a duration
Sample expression: 2 days from now (assuming **now** is 2000-10-03T18:19:06.270-04:00)
Returns: 2000-10-05T18:19:06.270-04:00

+
Description: unary plus operator
Sample expression: (+3 days)
Returns: 3 days
Note: the parentheses are required

-
Description: unary minus operator
Sample expression: (-50 hours)
Returns: -50 hours
Note: the parentheses are required

Binary operators:

+
Description: Adds a duration to another duration (returns a duration in seconds unless the duration specifiers are the same)
Sample expression: 340 days + 91 days
Returns: 431 days
Sample expression: 6 hours + 42 days
Returns: 3650400 seconds

-
Description: Subtracts a duration from another duration (returns a duration in seconds unless the duration specifiers are the same)
Sample expression: 340 days - 91 days
Returns: 249 days
Sample expression: 6 hours - 25 seconds
Returns: 21575 seconds

*
Description: Multiplies a duration by a number to obtain another duration. Order of arguments does not matter.
Sample expression: 40 days * 3
Returns: 120 days

Sample expression: 5 * 30 seconds
Returns: 150 seconds

/
Description: Divides a duration by a number to obtain another duration or divides a duration by a duration to obtain a number
Sample expression: 40 days / 2
Returns: 20 days
Sample expression: 2 minutes / 1 second
Returns: 120

<
Description: less than operator
Sample expression: 40 days < 26 days
Returns: false
Sample expression: 360 hours < 1 year
Returns: true

>
Description: greater than operator
Sample expression: 5 years > 12 months
Returns: true

<=
Description: less than or equal to operator
Sample expression: 26 minutes <= 26 minutes
Returns: true
Sample expression: 5 years <= 90 months
Returns: true

>=
Description: greater than or equal to operator
Sample expression: 9 years >= 9 years
Returns: true

= *or* ==
Description: equality operator
Sample expression: 3 days == 5 days
Returns: false

!= *or* <>
Description: inequality operator
Sample expression: 3 days != 5 days
Returns: true

List

A list can contain any of the basic operators listed on the first page (including lists). Operations supported on lists include membership checking, etc. Use of unsupported operators with lists is an error (causes a type mismatch exception to be raised).

Unary Operators

is list
Description: checks type of argument and returns true if it is a list
Sample expression: is list {{1, 2}, 3, "hey", 1999-03-04}
Returns: true
Sample expression: is list 567
Returns: false

first
Description: returns the first element in a list

Sample expression: first {2000-01-02T00:00:00, 24, 3, "hey", 1999-03-04}
Returns: 2000-01-02T00:00:00
Sample expression: first {{1, 2}, 3, "hey", 1999-03-04}
Returns: {1, 2}

last
Description: returns the last element in a list
Sample expression: last {2000-01-02T00:00:00, 24, 3, "hey", 1999-03-04}
Returns: 1999-03-04
Sample expression: last {{1, 2}, 3, "hey", "string"}
Returns: "string"

Binary Operators

is in
Description: checks whether first argument occurs in the list represented by the second argument
Sample expression: 2 is in {50, 99, 2, 3, "hey", 1999-03-04}
Returns: true
Sample expression: 55 is in {50, 99, 2, 3, "hey", 1999-03-04}
Returns: false

where
Description: the where operator is generally used to select values from a list, and has the form: "expr1 where expr2" (expr1 is usually a list, but can also be a value of any of the other basic types). The right argument to the where operator (expr2) is expected to be a logical expression, a list of extended boolean values, or **true**, **false**, or **unknown**. When the right argument is **true**, the left argument is returned unchanged. When it is **false** or **unknown**, an empty list is returned. When the right argument is a logical expression, it may make use of the keyword "it" to refer to the individual elements contained in the left hand side argument (when this is a list), or to refer to the non-list value that is the left hand side argument. The valid logical expressions that may appear on the right hand side of the where are:

is number it
is string it
is boolean it
is unknown it
is duration it
is time it
is list it

it < subexpr
subexpr < it
it <= subexpr
subexpr <= it
it > subexpr
subexpr > it
it >= subexpr
subexpr >= it
it == subexpr
subexpr == it
it != subexpr
subexpr != it
subexpr is in it
(where subexpr is a value of one of the basic types)

Sample expression: 1 where true
Returns: 1
Sample expression: 1 where false
Returns: {}
Sample expression: 1 where unknown
Returns: {}
Sample expression: 1 where {true, false, unknown, true, true}
Returns: {1, 1, 1}
Sample expression: {4,5,6,7,8,9,10} where it < 7

Returns: {4, 5, 6}
Sample expression: {4,5,6,7,8,9,10} where 7 < it
Returns: {8, 9, 10}
Sample expression: {4,5,6,7,8,9,10} where it <= 7
Returns: {4, 5, 6, 7}
Sample expression: {4,5,6,7,8,9,10} where 7 <= it
Returns: {7, 8, 9, 10}
Sample expression: {1,2,3,4,5,6,7} where it > 4
Returns: {5, 6, 7}
Sample expression: {1,2,3,4,5,6,7} where 4 > it
Returns: {1, 2, 3}
Sample expression: {1,2,3,4,5,6,7} where it >= 4
Returns: {4, 5, 6, 7}
Sample expression: {1,2,3,4,5,6,7} where 4 >= it
Returns: {1, 2, 3, 4}
Sample expression: {1,2,3,4,5,6,7} where it == 4
Returns: {4}
Sample expression: {1,2,3,4,5,6,7} where it != 4
Returns: {1, 2, 3, 5, 6, 7}
Sample expression: {{"CHF", "Mary", 1}, {"CHF", "Don", 2}, {"Angina", "Sam", 3}} where "CHF" is in it
Returns: {{"CHF", "Mary", 1}, {"CHF", "Don", 2}}
Sample expression: {{1,2}, 2, 4 hours, "hey", 1999-10-23, 3 days, "why", "one"} where 1 is in it
Returns: {{1, 2}}
Sample expression: interval[2,3] where 2 is in it
Returns: interval[2,3]
Sample expression: interval[2,3] where 9 is in it
Returns: {}
Sample expression: {{1,2}, 2, 3, 4, "hey", 1999-10-23, 3 days} where is number(it)
Returns: {2, 3, 4}
Sample expression: {"a", "b", 3 days, 4 hours} where is number(it)
Returns: {}
Sample expression: {{1,2}, 2, 3, 4, "hey", 1999-10-23, 3 days, "why", "one"} where is string(it)
Returns: {"hey", "why", "one"}
Sample expression: {{1,2}, 2, 3, 4} where is string(it)
Returns: {}
Sample expression: {{1,2}, 2, 4 hours, "hey", 1999-10-23, 3 days, "why", "one"} where is duration it
Returns: {4 hours, 3 days}
Sample expression: {{1,2}, 2, 4 hours, "hey", 1999-10-23, 3 days, "why", "one"} where is time(it)
Returns: {1999-10-23}
Sample expression: {{1,2}, 2, 4 hours, "hey", 1999-10-23, 3 days, "why", "one"} where is list(it)
Returns: {{1,2}}
Sample expression: {true, false, unknown, 1, 1999-03-04T05:00:00, "a"} where is boolean(it)
Returns: {true, false, unknown}
Sample expression: {true, false, unknown, 1, 1999-03-04T05:00:00, "a"} where is unknown(it)
Returns: {unknown}

Numeric Interval

Operations supported on numeric intervals include inclusion and overlap comparisons. The values appearing within a numeric interval specification are real numbers with the exception of the special keywords –infinity and infinity. An interval is specified by using the keyword "interval" followed by "[" (to represent an inclusive lower bound) or "(" (to represent a non-inclusive lower bound), and two comma-separated numbers followed by "]" (to represent an inclusive upper bound) or ")" (to represent a non-inclusive upper bound). The number specified as the lower bound must be less than or equal to the number specified as the upper bound. Use of unsupported operators with numerical interval values is an error (causes a type mismatch exception to be raised).

Binary Operators

is in
Description: checks whether first argument occurs in the interval represented by the second argument
Sample expression: 1 is in interval((-1), 5)
Returns: true

Sample expression: (-10) is in interval((-50), (-2))
Returns: true
Sample expression: 5 is in interval(5, 29]
Returns: false
Sample expression: 5 is in interval[5, 29]
Returns: true

overlaps
Description: checks whether two numeric intervals overlap
Sample expression: interval[5,29) overlaps interval[26, 900]
Returns: true
Sample expression: interval(1, 50) overlaps interval(1, 50)
Returns: true
Sample expression: interval[3,5) overlaps interval(5, 99]
Returns: false

Duration Interval

Operations supported on duration intervals include inclusion and overlap comparisons. The values appearing within a duration interval specification are durations. An interval is specified by using the keyword "interval" followed by "[" (to represent an inclusive lower bound) or "(" (to represent a non-inclusive lower bound), and two comma-separated durations followed by "]" (to represent an inclusive upper bound) or ")" (to represent a non-inclusive upper bound). The duration specified as the lower bound must be less than or equal to the duration specified as the upper bound. Use of unsupported operators with duration interval values is an error (causes a type mismatch exception to be raised).

Binary Operators

is in
Description: checks whether first argument occurs in the interval represented by the second argument
Sample expression: 1 day is in interval((-1 day), 5 days)
Returns: true
Sample expression: (-10 years) is in interval((-50 years), (-2 years))
Returns: true
Sample expression: 5 hours is in interval(5 hours, 29 days]
Returns: false
Sample expression: 5 hours is in interval[5 hours, 29 days]
Returns: true

overlaps
Description: checks whether two duration intervals overlap
Sample expression: interval[5 minutes, 29 minutes) overlaps interval[26 minutes, 900 minutes]
Returns: true
Sample expression: interval(1 month, 50 months) overlaps interval(1 month, 50 months)
Returns: true
Sample expression: interval[3 seconds, 5 minutes) overlaps interval(5 minutes, 99 hours]
Returns: false

Absolute Date and Time Interval

Operations supported on absolute date and time intervals include inclusion and overlap comparisons. The values appearing within an absolute date and time interval specification are absolute dates and times. An interval is specified by using the keyword "interval" followed by "[" (to represent an inclusive lower bound) or "(" (to represent a non-inclusive lower bound), and two comma-separated absolute date and time values followed by "]" (to represent an inclusive upper bound) or ")" (to represent a non-inclusive upper bound). The absolute date and time specified as the lower bound must occur before or equal the absolute date and time specified as the upper bound. Use of unsupported operators with absolute date and time interval values is an error (causes a type mismatch exception to be raised).

Binary Operators

is in

Description: checks whether first argument occurs in the interval represented by the second argument
Sample expression: 1999-03-04 is in interval(1998-10-12, 2000-02-05T05:00:00)

overlaps
Description: checks whether two absolute date and time intervals overlap
Sample expression: interval(1998-10-12, 2000-02-05T05:00:00) overlaps interval(1998-10-12, 2000-02-05T05:00:00)
Returns: true

GLIF3: The Evolution of a Guideline Representation Format

Mor Peleg, Ph.D.[1], Aziz A. Boxwala, M.B.B.S., Ph.D.[2], Omolola Ogunyemi, Ph.D.[2], Qing Zeng, Ph.D.[2], Samson Tu, M.S.[1], Ronilda Lacson, M.D.[2], Elmer Bernstam, M.D., M.S.E.[1], Nachman Ash, M.D.[2], Peter Mork, B.A.[1], Lucila Ohno-Machado, MD, Ph.D.[2], Edward H. Shortliffe, M.D., Ph.D.[3], Robert A. Greenes, M.D., Ph.D.[2]

[1]Stanford Medical Informatics, Stanford University School of Medicine, Stanford, CA
[2]Decision Systems Group, Harvard Medical School, Brigham & Women's Hospital, Boston, MA
[3]Department of Medical Informatics, Columbia University, New York, NY

The Guideline Interchange Format (GLIF) is a language for structured representation of guidelines. It was developed to facilitate sharing clinical guidelines. GLIF version 2 enabled modeling a guideline as a flowchart of structured steps, representing clinical actions and decisions. However, the attributes of structured constructs were defined as text strings that could not be parsed, and such guidelines could not be used for computer-based execution that requires automatic inference. GLIF3 is a new version of GLIF designed to support computer-based execution. GLIF3 builds upon the framework set by GLIF2 but augments it by introducing several new constructs and extending GLIF2 constructs to allow a more formal definition of decision criteria, action specifications and patient data. GLIF3 enables guideline encoding at three levels: a conceptual flowchart, a computable specification that can be verified for logical consistency and completeness, and an implementable specification that can be incorporated into particular institutional information systems.

1 Introduction

Clinical guidelines are potential tools for standardizing patient care to improve its quality and cost effectiveness. Unfortunately, guidelines have not always been successful at affecting clinician behavior. Structured, computer-interpretable guidelines can be delivered to the point of care in a way that enables decision support.[1] Such guidelines might also provide workflow management support, quality assurance evaluation, and simulation for educational purposes.[2]

There are several approaches to creating computer-interpretable guidelines that enable decision support. The PRO*forma* model assists patient care through active decision support and workflow management.[3] PRODIGY structures a guideline as a set of choices for the clinician,[4] and models patient scenarios that drive decision-making. PRESTIGE[5] uses a declarative approach to representing knowledge about the healthcare enterprise, the patient health record, and the protocol. The Asbru language[6] represents guidelines in a manner that includes explicit intentions of the guideline authors. The EON guideline model uses a combination of modeling primitives, such as various decision-making mechanisms, flow of control constructs, actions and activities, and a distinction between the normal case and its exceptions.[7] Arden syntax[8] is a language for creating and sharing medical knowledge in the form of independent units called medical logic modules (MLMs). Each MLM contains sufficient logic to make a single medical decision.

Creating clinical guidelines in computer-interpretable form takes significant effort. Thus, sharing them among developers and across institutions is desirable. However, there are many logistical obstacles to this goal. GLIF is a structured representation language of guidelines that was developed by the InterMed Collaboratory.[9] Its goals are to (1) enable viewing of GLIF-formatted guidelines by different software tools and (2) enable adapting the guidelines to a variety of local uses. Its goal is not to be a medium for translation from one guideline formalism to another.*

The objective of the GLIF specification is to provide a representation for guidelines that is: (a) precise and unambiguous; (b) human-readable; (c) computable, in the sense that the logic and sequence in guidelines specified in GLIF can be interpreted by computer; and (d) adaptable to different clinical information standards, thus facilitating guideline sharing.

2 Background

Version 2.0 of GLIF (GLIF2) was published in 1998,[9] and consisted of the GLIF object model and the GLIF

* In this sense, the word "interchange" in the expansion of the GLIF acronym is a misnomer.

syntax. The GLIF model, published in Interface Definition Language (IDL),[10] allowed the specification of a guideline as a flowchart of temporally ordered steps. These steps represented clinical decision and action steps. Concurrency was modeled using branch and synchronization steps. GLIF's guideline class also specified maintenance information (author, status, modification date, and version), the intention of the guideline, eligibility criteria, and didactics. The GLIF guideline instance syntax, which was based on a separately developed language, specified the format of text files, which contained GLIF-encoded guidelines. These files were used for sharing and interchange.

GLIF2 has been the basis for several implementations of guideline-based applications, including one in Brigham and Women's Hospital's BICS information system,[11] and web-based applications for driving clinical consultations.[2] However, GLIF2 has certain deficiencies that limit its usability. As a result, non-standard extensions had been made to GLIF2 to implement the above applications. The deficiencies are:

1. GLIF2 does not specify how to structure important attributes of guideline steps, such as data and action names and logical condition expressions. Values of most attributes are specified simply as text strings. Thus, such guidelines cannot be used for automatic inference.
2. Integrating GLIF2 guidelines with heterogeneous clinical systems is difficult, as GLIF2 lacks features for mapping patient data references to entries in the electronic medical record.
3. GLIF2's decision model is limited. Decisions are either specified in a conditional step that models if-then-else semantics, or in a branch step for which no preference among the alternatives can be expressed.
4. GLIF2 provides only a limited set of low-level constructs. Important concepts such as those for describing iteration, patient-state, exception conditions, and events are lacking.
5. GLIF2 uses subguidelines to manage complexity in guideline flowcharts. These subguidelines can be used to expand action steps. However, because GLIF2's set of constructs is limited, GLIF2 guidelines tend to be cumbersome, even if they do use subguidelines.
6. The branch step can be used both for representing concurrent execution of multiple actions and for making selection among a set of alternatives. Thus, its semantics are a mixture of concurrency and decision-making.

This paper presents GLIF3, an evolving revision of GLIF that attempts to overcome several of GLIF2's limitations.**Overview of GLIF3**

GLIF3 enables guideline specification at three levels: a conceptual GLIF flowchart, a computable/parsable specification and an implementable specification. In addition, GLIF3 introduces substantive changes to GLIF2's object model and syntax. GLIF3 is intended to be sufficiently expressive to support specification of guidelines that differ in these ways: (1) their medical purposes (e.g., screening, disease management); (2) their intended uses (reference, patient management, and education); (3) the intended users (e.g., physician, patient); and (4) their utilization sites (e.g., ICU, out of hospital)[12]. We tried to avoid overlap in the functionality of different GLIF3 constructs, and not to enable a single GLIF construct to model two different guideline situations. For example, the branch step is no longer used to represent decision choices.).

3.1 Guideline Abstraction Levels

GLIF3 enables modeling of guidelines at three levels of abstraction:

A. Conceptual level. Guidelines at this level are represented as flowcharts that can be used for browsing, through guideline viewing programs. However, these guidelines cannot be used for computation in providing decision support.

B. Computable level. Guidelines at this level may be verified for logical consistency and completeness. Expression syntax, definitions of patient data items and clinical actions, and flow of the algorithm are specified at this level.

C. Implementable level. At this level, guidelines are appropriate for incorporation into particular institutional information system environments. Thus, these guidelines may incorporate non-sharable elements.

Figure 1 shows part of the conceptual specification of a guideline for management of stable angina.[13]**Changes in the object model**

The object model for GLIF3 defines new constructs and further structures GLIF2 constructs.

Representation in UML

The GLIF3 model is described using Unified Modeling Language (UML) class diagrams[14]. Additional constraints on represented concepts are being specified in the Object Constraint Language (OCL), a part of the UML standard.[14]

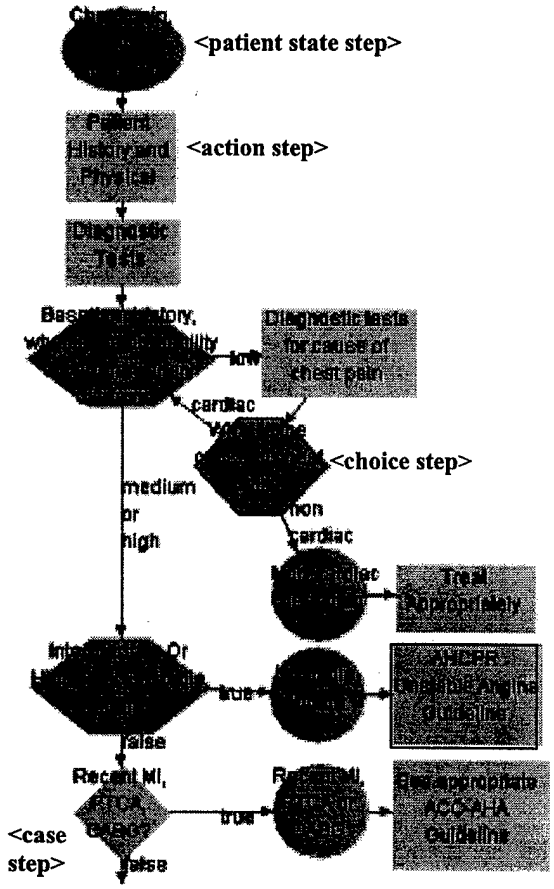


Figure 1. Conceptual flowchart specification of part of a stable angina guideline.

Support for managing complexity of guidelines

In comparison with GLIF2, GLIF3 more fully defines a mechanism for specifying guideline steps recursively through the nesting of subguidelines in action and decision steps. For example, *AHCPR Unstable Angina Guideline*, shown in Figure 1 as an action step, can be expanded by zooming, through the nesting mechanism, to show its details in the form of another flowchart diagram. Because nesting allows grouping of parts of a guideline into modular units (subguidelines), it is a mechanism that allows guideline parts to be reused. Furthermore, the modularity resulting from nesting permits adaptation of a guideline to a specific institution by replacing or elaborating upon specific sections of the guideline. For example, an action specified at a high-level may be replaced with a detailed procedure.

A new feature in GLIF3 is the *macro step*. Like Visual Basic, Object Linking and Embedding Custom Control (OCX), and Java Beans, a macro step is a special class with attributes that define information needed to instantiate a set of underlying GLIF steps. For example, as shown in Figure 2a, an MLM can be described using a pattern of GLIF components: a decision step that contains a *criterion* (logic slot) and is triggered by *events* (evoke slot), followed by an action step that include *action specifications* (action slot). Macro steps benefit authoring, visual understanding, and execution of guidelines. They also enable declarative specification of a procedural pattern that is realized by a flowchart of guideline steps.

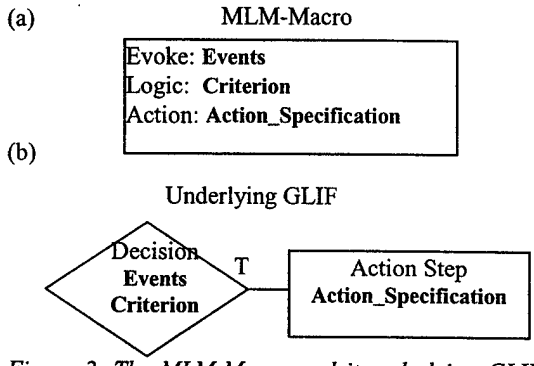


Figure 2. The MLM-Macro and it underlying GLIF pattern. (a) MLM-Macro; (b) underlying GLIF

In GLIF3, we added a capability that provides multiple views of the same guideline. Since different users may be interested in different parts of a large, complex guideline, differential display capability is supported. This capability is provided through the use of filters that collapse segments of the guideline into a default view of the guideline customized to a given user, situation, etc.

Expression specification

We added to GLIF3 a structured grammar for specifying expressions and criteria. The grammar can specify logical criteria, numerical expressions, temporal expressions, and text string operations. It is a superset of the Arden Syntax logic grammar,[15] and adds new operators such as "is a", "overlaps", "xor", "from now", "is unknown" and "at least k of ...".

Domain ontology support

In GLIF2, an Action Specification contained a Patient Data class that textually defined patient data items.

GLIF3 facilitates using of standard medical vocabularies and integrating shared guidelines into clinical information systems environments via a layered approach for referencing clinical terms. The *core GLIF* layer provides a standard interface to all medical data and concepts that may be represented and referenced by GLIF. The interface views all data items as being literals (constants) or variables. Each data item may

refer to a concept that is defined by the two other domain ontology layers. This approach enables each data item to contain specific relevant attributes. The *Reference Information Model* (RIM) *layer* provides a semantic hierarchy for medical concepts, and allows attribute specification for each class of medical data. Different RIMs, such as the HL7 RIM, may be used in different guidelines.

The *medical knowledge layer* contains a term dictionary (e.g., UMLS) and can provide access to medical knowledge bases. It can provide more specific information about medical concepts and their interrelationships. With such knowledge, we can examine the correctness of criteria and action specifications by performing range checks and semantic checks (e.g., a body-part has no "timestamp" attribute).

Flexible decision model

GLIF3 provides a flexible decision model through a hierarchy of decision step classes. This decision hierarchy distinguishes between decision steps that can be automated (*case steps*) and ones that have to be made by a physician or other health worker and cannot be automated (*choice steps*). Examples of case and choice steps are shown in Figure 1. The decision hierarchy can be extended in the future to model decisions that consider uncertainty or patient preferences. The hierarchy might be extended to support different decision models.

Extended action specification model

The action specification model has been extended to include two types of actions: (1) guideline-flow-relevant actions, such as calling of a sub-guideline, or computing values for data; and (2) clinically relevant actions, such as making recommendations. Clinically relevant actions reference the domain ontology for representations of clinical concepts such as prescriptions, laboratory test orders, or referrals.

Other new concepts

Representations for several new concepts were added to GLIF3. They include specifications for the following:

- Describing *Iterations* and conditions that control the iteration flow.
- Describing *Events* and triggering of guideline steps by events.
- Describing *Exceptions* in guideline flow and associated exception-handling mechanisms.
- Representing *Patient-State* as another kind of guideline step (a node in the flowchart), in addition to the existing action, decision, branch, and synchronization steps. A patient-state step serves as an entry point into the guideline and as a label summarizing the patient's condition. The patient-state step has a precondition attribute. A patient whose state matches the precondition criterion is potentially in that state. Figure 1 shows several patient state steps.
- A *Keyword Didactic* for adding keywords to a variety of constructs in guidelines.

Corrections to branch and synchronization step

The branch step has been modified to remove redundancy between it and the decision step. In addition, the branch and synchronization steps have been modified to remove redundancy in descriptions of parallel pathways in the guideline flowchart.

3.3 Changes in the GLIF syntax

XML-based syntax

The proprietary ODIF-based syntax[16] in GLIF2 is being replaced with an RDF-based syntax[17] syntax that relies on XML for serialization. We have developed a schema for the syntax.

4 Discussion

GLIF is an effort to create a community-supported guideline representation methodology that will facilitate sharing of computer-interpretable clinical guidelines. It was developed through a collaboration of a number of institutions, including Stanford Medical Informatics; the Decision Systems Group of Brigham & Women's Hospital, Harvard Medical School; the Department of Medical Informatics at Columbia University; and the Center for Medical Education at McGill University. The Laboratory for Computer Science at Massachusetts General Hospital, participated in the development of GLIF2. GLIF3 tries to leverage the years of effort that have gone into the development of other existing methodologies. Like EON[7], GLIF models a clinical guideline as a flowchart. GLIF3 includes the patient-state step that is similar in functionality to scenarios, which are used in PRODIGY[4]. GLIF3 also uses a superset of Arden Syntax[8] for expressing decision criteria and supports the MLM-macro that can be used to map GLIF-encoded guidelines into MLMs.

GLIF3 is evolving very rapidly. More work still needs to be done on the specification of its domain ontology. We are currently specifying several clinical guidelines, at the three abstraction levels, in order to evaluate GLIF3. To solicit comments from the com-

munity, the current GLIF3 specification is published on the Internet at http://www.glif.org/glif3_info.html.

Future versions of GLIF will explore structured representations for (1) specifying goals of guideline steps,[6] (2) probabilistic models for decision-making,[18] and (3) incorporation of patient preferences in decision steps.[19]

We are developing software tools for authoring, verifying, viewing, distributing, and executing guidelines. These tools are being implemented in Java to provide portability and use over the Internet.

5 Acknowledgements

Supported in part by Grant LM06594 from the National Library of Medicine and by the Telemedicine and Advanced Technology Research Center, U.S. Army Medical Research and Material Command.

Bibliography

1. Lobach DF, Hammond WE. Computerized decision support based on a clinical practice guideline improves compliance with care standards. Am J Med 1997;102(1):89-98.

2. Boxwala AA, Greenes RA, Deibel SR. Architecture for a multipurpose guideline execution engine. In: Proc AMIA Symp; 1999. p. 701-705.

3. Fox J, Rahmanzadeh A. Disseminating medical knowledge: the PROforma approach. Artificial Intelligence in Medicine 1998;14:157-181.

4. Sugden B, Purves IN, Booth N, Sowerby M. The PRODIGY Project - the Interactive Development of the Release One Model. In: AMIA Symp; 1999. p. 359-363.

5. Gordon C, Veloso M. Guidelines in Healthcare: the experience of the Prestige project. In: Medical Informatics Europe; 1999; Ljubljana, Slovenia; 1999. p. 733-738.

6. Shahar Y, Miksch S, Johnson P. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. Artificial Intelligence in Medicine 1998;14:29-51.

7. Tu SW, Musen MA. A Flexible Approach to Guideline Modeling. In: AMIA Symp; 1999. p. 420-424.

8. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res 1994; 27(4):291-324.

9. Ohno-Machado L, Gennari JH, Murphy S, Jain NL, Tu SW, Oliver DE, et al. The GuideLine Interchange Format: A Model for Representing Guidelines. Journal of the American Medical Informatics Association 1998;5(4):357-372.

10. Object Management Group. The Common Object Request Broker: Architecture and Specification; 1999. Report No.: OMG Document Number 91.12.1.

11. Zielstorff RD, Teich JM, Paterno MD, Segal M, Kuperman GJ, Hiltz FL, et al. P-CAPE: a high-level tool for entering and processing clinical practice guidelines. Partners Computerized Algorithm and Editor. Proc Amia Symp 1998:478-82.

12. Bernstam E, Ash N, , Peleg M, Tu S, Boxwala AA, et al. Guideline classification to assist modeling, authoring, implementation and retrieval. In: Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium; Los Angeles, CA,: American Medical Informatics Association; 2000.

13. American College of Cardiology/American Heart Association/American College of Physicians-American Society of Internal Medicine. Guidelines for the Management of Patients with chronic Stable Angina. J Am Col Cardiol 1999;33:2092-2197.

14. Object Management Group. Unified Modeling Language (UML) Specification. In: http://www.rational.com/uml/index.jtmpl. version 1.3 ed; 1999.

15. Clinical Decision Support & Arden Syntax Technical Committee of HL7, inventor Arden Syntax for Medical Logic Systems, version 2.0. Draft revision. USA. 1999 July 7, 1999.

16. Pattison-Gordon E. ODIF: Object Data Interchange Format. Boston, MA: Decision Systems Group, Brigham and Women's Hospital; 1996. Report No.: DSG-96-04.

17. W3C. Extensible Markup Language (XML). In: http://www.w3.org/XML/; 2000.

18. Aliferis CF, Miller RA. On the heuristic nature of medical decision-support systems. Methods of Information in Medicine 1995;34:5-14.

19. Barratt A, Irwig L, Glasziou P, Cumming RG, Raffle A, Hicks N, et al. User's guide to the medical literature: How to use guidelines and recommendations about screening. JAMA 1999;281(21):2029-34.

Finding Appropriate Clinical Trials: Evaluating Encoded Eligibility Criteria with Incomplete Data

Nachman Ash, M.D.,M.S. [1,2], Omolola Ogunyemi, Ph.D. [1], Qing Zeng, Ph.D. [1], Lucila Ohno-Machado, M.D., Ph.D. [1,2]

[1]Decision Systems Group, Brigham and Women's Hospital, Harvard Medical School
[2]Division of Health Sciences and Technology, Harvard Medical School and Massachusetts Institute of Technology, Boston, MA

ABSTRACT

We describe our work on creating a system that selects appropriate clinical trials by automating the evaluation of eligibility criteria. We developed a data model of eligibility for breast cancer clinical trials, upon which the criteria were encoded. Standard vocabularies are utilized to represent concepts used in the system, and retrieve their hierarchical relationships. The system incorporates Bayesian networks to handle missing patient information. Protocols are ranked by the belief that the patient is eligible for each of them. In a preliminary evaluation, we found good agreement (kappa 0.86) between the system and an independent physician in selection of protocols, but poor agreement (kappa 0.24) in protocol ranking. We conclude that our approach is feasible, and potentially useful in assisting both physicians and patients in the task of selecting appropriate trials.

INTRODUCTION

The important role of informatics in all stages of clinical trials is well established, encompassing patient accrual, protocol management and evaluation of results. The National Cancer Institute (NCI) plans to create a web enabled Cancer Informatics Infrastructure (CII) through which all aspects of clinical trials will be accessible[1,2]. Silva describes one of the major aspects of this vision: "...by using their computer, patients and their oncologists can find, for the patient's specific cancer, the best treatments and clinical trials" [1].

While information regarding clinical trials is currently easily accessible via the web[3], the task of finding appropriate clinical trials for a specific patient is tedious, requiring the evaluation of hundreds of eligibility criteria. Physicians often do not have enough time to perform this task, while patients may lack the knowledge and skills required.

Several methodologies were developed for evaluating patients' eligibility for clinical trials[4-8]. All of them aimed at improving the accrual of patients to specific trials. Ohno-Machado et al took a different approach by focusing on the patient. Their system allows the patient or her provider to obtain a ranked list of clinical trials for which the patient is likely to be eligible[9].

In this paper we present our extension to their work. We address the major concerns raised in that study: (1) the authors were able to encode only about 50% of the criteria, ignoring the most complex ones, and (2) they used a deterministic algorithm that did not take into account missing patient data. We designed an object oriented data model, and introduced the use of concepts and relationships from standard medical vocabularies to facilitate the encoding of complex criteria. In addition, our system makes use of Bayesian networks to handle the problem of missing patient data. We also present a preliminary evaluation of the system.

MATERIALS AND METHODS

Source of protocols. The clinical trial protocols were taken from NCI's Physician Data Query (PDQ) database[10]. We focused on phase II and phase III trials for the treatment of metastatic or recurrent breast cancer in women (see [9] for more details). Seventy-nine protocols have been retrieved using these criteria as of February 2001.

Implementation. We redesigned our system based on the following principles (Figure 1):

- Medical knowledge is encapsulated in an object-oriented data model.
- Concepts are represented using standard vocabularies.
- Eligibility criteria are encoded in a logical expression language derived from Arden syntax.
- Encoded eligibility criteria are stored in a database for reuse and future sharing.
- Bayesian networks are incorporated into the system's evaluation process for inferring missing patient data.
- Evaluated protocols are ranked by the likelihood that the patient is eligible for each of them.
- The system has a platform-independent implementation based on Java.

Knowledge representation. The data model's structure is based on analysis of the breast cancer protocols and the Common Data Elements (CDE) of breast cancer clinical trials developed by NCI[1]. The model captures the data items in these protocols, their temporal aspects, and relationships among them. It is the basis for storing the patient data and checking for allowed values and inconsistencies.

The concepts used in the system are represented using standard vocabularies in the UMLS. We chose to use MeSH and PDQ, which contain the relevant concepts, and capture appropriate hierarchical relationships.

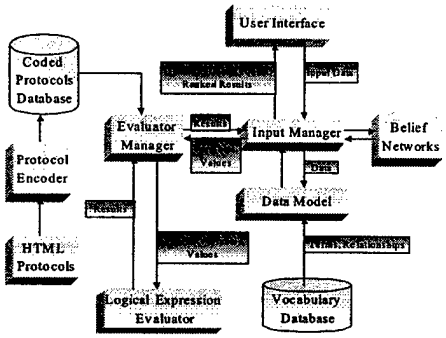


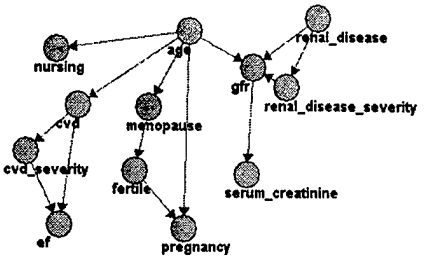
Figure 1: System architecture.

Encoding the protocols. Currently, the first 10 protocols out of the 79 retrieved from the PDQ database have been encoded. The HTML version of each protocol was automatically parsed to extract the textual eligibility criteria. These criteria were encoded manually (by the first author) using a variation of the Guideline Expression Language (GEL)[11]. The language contains the expressions used to retrieve data from the object model (based on pre-defined functions) as well as logical expressions.

We created a special editor for encoding the criteria. It lets the user check the syntax of an expression for correctness, verify the legitimacy of variables' names used in the expression, and assess whether the terms used in the expression map to concepts in the UMLS. When a criterion in a protocol is identical to a previously encoded criterion from a different protocol, its GEL-based encoding is retrieved automatically from the database. The time taken to encode each criterion is measured and saved for analysis.

Inferring missing data. We incorporated Bayesian networks into the new system to infer missing data based on population-based probabilities of patients' characteristics. Some of the probabilities were obtained or calculated from the medical literature and known statistical databases[12]. The first author estimated others based on his medical knowledge. Since the estimated probabilities are not optimal, we plan to augment them by using relevant patient data, as it becomes available, as suggested by Neapolitan[13].

The Bayesian network structure is based on causal and associational relationships identified from the data model and the common data items used in the protocols. Currently, it has 31 nodes and contains 4 separate directed acyclic graphs representing age-related items (Figure 2), liver function tests, white blood cell counts and pulmonary function tests. The software used for creating the network is JavaBayes[14].



ef - Ejection Fraction cvd - CardioVascular Disease
gfr - Glomerular Filtration Rate

Figure 2: Directional graph of one of the Bayesian networks used in the system.

Evaluating criteria. Encoded criteria are evaluated using a three-valued logic (true, false, unknown) by a parser and interpreter created for GEL.

Ranking the protocols. Protocols for which all eligibility criteria evaluate to "true" given patient's data are ranked highest. Those that contain at least one criterion that evaluates to "false" are filtered out. The remaining protocols, containing at least one criterion that evaluates to "unknown", are ranked according to the belief that the patient is eligible for each of them. The ranking algorithm uses heuristics that take into account the following:

- Number of unknown criteria.
- A discriminatory score of each unknown criterion. An inclusion criterion that is probably true for most patients gets a different score than one that is probably true for only a small subset of patients. For example, "age greater than 18" is more inclusive than "age greater than 65", and therefore if the age of the patient is unknown, there is a greater chance that she meets the first criterion.
- Number of "inferred criteria" (criteria that originally evaluate to "unknown", and later to "true" or "false" based on inferred patient data).
- The evaluation result of the inferred criteria. A protocol containing a criterion that evaluates to false using inferred data is not filtered out, but rather gets a score that will rank it lower.

The final score of a protocol is given on a scale from 1 (definitely inappropriate) to 5 (definitely appropriate).

Criterion Difficulty	Number of Criteria	Average Encoding Time (Min)
Automatic Coding	18	≈ 0
Trivial	8	1.47
Easy	35	3.52
Difficult	9	11.12
Complex	5	28.12
Very Complex	2	36.80

Table 1: Average encoding time of 77 criteria stratified by difficulty of encoding.

Evaluation. Patient data were abstracted from medical records of patients with active metastatic or recurrent breast cancer, who were consecutively hospitalized during 1995 at the Brigham and Women's Hospital, Boston, Massachusetts. Forty-three data items were examined for each patient (items related to patient characteristics, disease characteristics, past treatment, other diseases and test results). The data collection process was separate from the protocol encoding process.

An independent physician (oncologist, but not a breast cancer specialist) evaluated the appropriateness of the protocols for each of the patients, grading the protocols as described above (on a 1-5 scale) and ranking them. The physician was given the patients' data in a short narrative description, and the full abstracts of the protocols as downloaded from NCI's CancerNet web site.

Statistical analysis. The agreement of the system and the physician on selection and ranking of protocols was calculated using the kappa and weighted kappa statistics[15].

RESULTS

Encoding process. We encoded 10 protocols each containing 20 - 41 eligibility criteria (mean 27.2). 228 criteria out of 272 (83.8%) were unique. We were able to encode 269 criteria (98.9%). For two of the three uncoded criteria ("no prisoners" and a request for a specific geographic location), the model could be improved to capture the necessary knowledge. The third ("No other concurrent medical or psychological condition that would preclude study compliance") was difficult to encode for automatic evaluation. A total of 39 other criteria (14.3%) did not represent their text version with 100% accuracy (e.g., "No medical or psychiatric condition that would increase risk" was encoded as "No severe medical or psychiatric condition". Since assessment of risk is subjective, it is difficult to encode for computation).

A significant number (30.3%) of the encoded criteria were lengthy (> 255 characters), suggesting the proportion of more complex criteria.

Table 1 presents the encoding time of 77 criteria from the last 3 protocols. The average encoding time was 5.88 minutes (median 2.1 minute). Therefore, encoding an average sized protocol may take about 3 hours.

Data Item	No. of patients(percent)
Stage:	
Stage IV	5 (25%)
Stage IIIb	5 (25%)
Unknown	10(50%)
Histology:	
Invasive Ductal Ca.	1 (5%)
Unknown	19 (95%)
Confirmed Histology/Cytology	17 (85%)
Measurable/Evaluable Disease	14 (70%)
Menopausal Status	
Postmenopausal	5 (25%)
Premenopausal	8 (40%)
Unknown	7 (35%)
Known Metastases	11 (55%)
Recurrent Disease	3 (15%)
Locally Advanced Disease	8 (40%)
Known Lymph Node Involvement	9 (45%)
Other Diseases	
Hypertension	3 (15%)
NIDDM*	1 (5%)
Asthma	1 (5 %)
Past Treatment	
Chemotherapy	16 (80%)
Radiotherapy	6 (30%)
Biotherapy	8 (40%)
Hormonal therapy	7 (35%)
Surgery	7 (35%)

*Non Insulin Dependent Diabetes Mellitus

Table 2: Patient characteristics.

Preliminary system evaluation. Data from records of 20 patients with metastatic, locally invasive, and recurrent breast cancer were collected. In average, about 25% of the 43 data items collected for each patient had missing values. Age distribution was 25-71 years (mean 44.4). Other patient characteristics are shown in table 2.

The process of protocol selection for these 20 patients involved 5400 evaluations of 272 criteria. Table 3 presents the evaluation results of these criteria.

The system selected 1 - 9 protocols per patient (3.05 protocols on average, overall 61 protocols were selected for 20 patients). None of the protocols evaluated to a score of 5 (definitely eligible) or 4 (probably eligible), 25 were graded 3 (possibly eligible), and 36 were graded 2 (low probability for eligibility).

Evaluation Result	Criteria Number (percent)
TRUE	2287 (42.04%)
FALSE	223 (4.10%)
UNKNOWN	2930 (53.86%)
true (inferred)	543 (9.98%)
false (inferred)	39 (0.72%)
unknown	2348 (43.16%)

Table 3: Results of 5440 evaluations of eligibility criteria.

The system's results were compared to the physician's selection of protocols in two aspects: the agreement on whether the patient is eligible for each protocol (Table 4), and the agreement on protocol ranking for each patient. The kappa statistic for appropriateness of protocols was 0.86 (95% CI 0.72 - 1.00). For 11 out of 20 patients (55%) both the system and the physician ranked the same protocol as first (kappa 0.37). The weighted kappa for ranking the protocols was 0.24.

		Physician Selection		
		Selected	Not Selected	Sum
System	Selected	59	2	61
	Not Selected	10	129	139
	Sum	69	131	200

Table 4: Selection of protocols by the system compared to a physician's selection.

DISCUSSION

Our results show that encoding and automatically evaluating eligibility criteria to find appropriate clinical trials for a specific patient is feasible.

We were able to encode 98.9% of the criteria, as compared to about 50% in the previous version of the system. This is the result of using an elaborated data model and standard vocabularies. Yet, we had difficulty encoding some of the ambiguous criteria that must involve human judgment.

The encoding language requires familiarity with the data model. Nevertheless, we share the vision that authors of clinical trial protocols will encode the criteria by themselves[16], and believe that it will be possible if a library of encoded criteria is provided.

Using terms from standard vocabularies is powerful in many aspects. It enabled us to simplify the data model and make it scalable. Thus, although the system is currently restricted to breast cancer protocols, it may be expandable to other domains.

Different approaches have been used in the past to handle missing data in evaluating eligibility for clinical trials. Tu[17] suggested combining qualitative and probabilistic approaches, while Papaconstantinou[8] used a probabilistic system in which the whole protocol is translated into a Bayesian network.

Our approach is somewhat different from the two mentioned in combining deterministic and probabilistic methods for inferring missing values. Deterministic inference involved, for example, deducing that a patient with metastases has a stage IV disease. Table 2 shows that 11 of the patients were known to have metastasis, while only 5 were known to have stage IV disease. Our system infers that patients with known metastases have stage IV disease (and vice versa). This kind of inference is crucial for appropriate selection of protocols, since we allow filtering out of protocols based on these inferred items.

In addition we modeled several small independent Bayesian networks that capture dependencies among different data items (e.g. liver diseases and liver function tests). Each variable in the network, which has a missing value, due to lack of patient data, has its value inferred by the Bayesian network. Evaluation of criteria that make use of these inferred values produces a qualitative estimate that the patient meets these criteria. Using small networks makes it relatively easy to build and expand them, and it might be simpler to find the needed prior and conditional probabilities that populate them.

The impact of the Bayesian networks was rather small. Although up to 20% of missing variables were inferred, it didn't have a major effect on ranking protocols (the system ranked the protocols the same when used without the Bayesian networks). We believe that this is the consequence of the paucity of patient data (as shown in table 3, more than 50% of criteria were evaluated to unknown). The impact of the Bayesian network will probably be higher if more data are entered into the system.

Our results show fairly good agreement between the system and a physician on protocol selection. It can potentially be a reliable means to select protocols. In this way, it can save practitioners a lot of time since many protocols are filtered out (more than 2/3 in our evaluation). We envision that such a system can be incorporated into the CII project of the NCI.

The agreement on ranking the protocols was much lower. Since the ranking process can be more subjective, these results are not surprising. As we lack a gold standard, we cannot decide which (system's or physician's) ranking is better. We plan to continue investigating this issue.

The study has several limitations that will direct our future work. Independent users will test the coding process, so we can learn about the applicability of the process.

The small number of encoded protocols limited the evaluation of the system. On the other hand, a larger number would probably be less manageable for evaluation by physicians. Since our conclusions are currently based on one physician, we plan to recruit more physicians to evaluate the protocols, some of whom will be domain experts and some general practitioners.

We plan to collect more data items, in particular temporal data, in order to test other aspects of the coded criteria. Finally, we plan to complete the user interface and evaluate the use of the system by practitioners and patients.

ACKNOWLEDGEMENTS

This project was funded by contract 34078PP1024 from the Massachusetts Department of Public Health, and grant DAMD 17-98-1-8093 from the Department of the army. Dr. Ash is also supported by the American Physicians Fellowship Committee (APFC) of the Israel Medical Association, and by Dr. Arthur Holstein's Fund of the American Jewish Joint Distribution Committee. We would like to thank Dr. Ronilda Lacson and Ms. Debra Della Torre for collecting the patient data.

REFERENCES

1. Silva, J and Wittes R, Role of clinical trials informatics in the NCI's cancer informatics infrastructure.
   Proc AMIA Symp, 1999;p. 950-4.
2. Silva, JS, Fighting cancer in the information age. An architecture for national scale clinical trials. MD Comput, 1999;16:43-4.
3. McCray, AT, Better Access to Information about Clinical Trials.
   Ann Intern Med, 2000;133:609-614.
4. Butte, AJ, Weinstein DA, and Kohane IS, Enrolling patients into clinical trials faster using real time Recruiting.
   Proc AMIA Symp, 2000; p. 111-5.
5. Carlson, RW, Tu, SW, Lane, NM, et al., Computer-based screening of patients with HIV/AIDS for clinical-trial eligibility.
   Online J Curr Clin Trials, 1995;Doc No 179
6. Breitfeld, PP, Weisburd, M, Overhage, JM, Sledge, G, and Tierney, WM, Pilot study of a point-of-use decision support tool for cancer clinical trials eligibility. J Am Med Inform Assoc, 1999;6:466-77.
7. Gennari, JH and Reddy M, Participatory design and an eligibility screening tool.
   Proc AMIA Symp, 2000:p. 290-4.
8. Papaconstantinou, CG, Theocharous G, and Mahadevan S, An expert system for assigning patients into clinical trials based on Bayesian networks. J Med Syst, 1998;22:189-202.
9. Ohno-Machado, L, Wang, SJ, Mar, P and Boxwala, AA, Decision support for clinical trial eligibility determination in breast cancer.
   Proc AMIA Symp, 1999:p. 340-4.
10. Physician Data Query (PDQ) - National Cancer Institute comprehensive database that contains cancer information summaries.
    http://cancernet.nci.nih.gov/trialsrch.shtml, Accessed 2/1/2001.
11. Ogunyemi O, The Guideline Expression Language (GEL) User's guide, Technical Report, DSG-TR-2000-001, 2000, Decision Systems Group, Boston, MA.
12. Ries LAG, Wingo PA., Miller DS, et al., The Annual Report to the Nation on the Status of Cancer, 1973-1997,
    Cancer, 2000.;88:2398-2424.
13. Neapolitan RE, Probabilistic reasoning in expert systems: Theory and algorithms.
    Wiley, New York, 1990.
14. Cozman, FG, JavaBayes - a system that handles Bayesian networks implemented in Java.
    http://www.cs.cmu.edu/~javabayes/Home/, accessed 2/1/2001.
15. Altman, D, Practical Statistics for Medical Research. CRC press, LLC, 1991:p. 403-409.
16. Rubin, DL, Gennari, JH,Srinivas, S. et al., Tool support for authoring eligibility criteria for cancer trials.
    Proc AMIA Symp, 1999: p. 369-73.
17. Tu, SW, Kemper, CA, Lane, NM, Carlson, RW and Musen, MA, A methodology for determining patients' eligibility for clinical trials. Methods Inf Med, 1993;32:317-25.